AN IFC-BASED FRAMEWORK FOR OPTIMIZING LEVEL OF PREFABRICATION IN INDUSTRIALIZED BUILDING SYSTEMS

ALIREZA KHALILI

(B. Eng, M.Sc, Sharif University of Technology)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2013

Acknowledgments

In the name of GOD the most compassionate the most merciful. This research would not have been possible without the help of numerous people particularly the support and understanding that my wife and my family gave me throughout the PhD candidature. Without their support I would not have sustained through the stress and frustration of this research.

I would like to express my deepest gratitude to my supervisor, Associate Professor David K. H. Chua, for his advice, support, patience, and encouragement throughout the course of this research. It is not often that one finds an advisor who is always energetic and active in both academic and consultancy fields. I am also grateful that despite his busy schedule he managed to allocate time to read and comment critically on my papers and thesis.

I would like to express my special thanks to Mr. Ernest Abbott, who spent a lot of time, energy, and patience to help proofread the thesis draft. I also enjoyed the times when we discussed many research topics on IFC, programming, and Graph analysis. I am also grateful to my officemates Ms. Nguyen Thi Qui, and Mr. Meghdad Attarzadeh who provided several valuable suggestions for my research.

My thanks also go to the management staff of the Mahan Company for the information provided for the case study and all their technical support.

My genuine acknowledgement is given to National University of Singapore for providing research scholarship. I am also very grateful to all the individuals that helped the research in one way or another, but they were not named due to space constraints or their request for anonymity. I also hope that the ideas proposed in this thesis will bring about improvement to the Industrialized Building Systems and prefabrication.

I dedicate this thesis to my wife Mona, who has brought much love into my life, my mother, father and brother for their unwavering support.

Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

ALIREZA KHALILI

Table of Contents

Abstract

Over the past six decades Industrialized Building Systems (IBSs) have been adopted by the construction industry all over the world to improve overall productivity in construction projects. The benefits to the construction industry can be improved by adopting prefabrication, which utilizes advanced manufacturing systems. These systems are more efficient and generally increase the construction quality and customer satisfaction. Prefabrication is one form of industrialization in construction industry that may have various levels of standardization. Standardization provides faster production, lower cost, and more efficient assembly of elements due to uniform dimensions that eliminate costly and time-consuming custom-made applications while still allowing multiple configurations. A greater benefit could be achieved with a higher portion of offsite works.

This dissertation presents a framework on the incorporation of spatial relationships of building elements and constructability analysis in resource planning and scheduling of prefabrication using artificial intelligence techniques. This framework aims to move beyond traditional prefabrication method in which building elements are produced individually towards configuring higher level of prefabrication. Level of prefabrication is coined to emphasize the importance of amount of off-site work in Industrialized Building Systems.

In this regard, the proposed framework which is called Component Configuration of Prefabricated Structures (CoCoPS) configures the grouping of elements into higher level components in order to minimize the total number of components so as to reduce transportation and installation costs and at the same time to maximize the number of identical components due to the economy of scale in terms of mould fabrication cost.

The knowledge embodied in parametric modeling serves as a semantic data model as well as a data exchange platform for configuring building elements and grouping them for a higher degree of prefabrication. The framework extracts semantic, geometric, and geometrical properties of building elements from the Industrial Foundation Classes (IFC). Topological relationships among building elements are deduced so that all possible configurations of building elements are generated. A rule-based database is utilized to check constructability of the generated configuration in terms of design, production, transportation, and installation.

Keywords: Industrialized Building System (IBS), Prefabrication, Standardization, CoCoPS, GDM, Resource Optimization, Artificial Intelligence

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

Over the past six decades Industrialized Building Systems (IBSs) have been adopted by the construction industry all over the world to improve overall productivity in construction projects. IBS aims to increase buildability of designs and quality of the construction products and to decrease the dependency on low-skilled workers. Various designs, multiple products, multi-stage operations, advanced production technology, fluctuating demand, and limited resources are a few characteristics of today's industrialized building industry. The degree of flexibility in design and production and the ease of adaptability to changing market scenarios force this industry to apply new methods in its planning and scheduling.

Current practices in the implementation of IBS are mainly focused on prefabricated products and planning for production. This may lead to inefficient resource utilization, higher cost and/or time. Literature review in the area of industrialization in the building industry reveals that a greater value would be added to a project by increasing the degree of design-production integration and spatial integration of individual building elements to components and modules. The implementation of industrialization in buildings cannot be examined today without considering the impact of information technologies on the design, production, and assembling onsite of prefabricated elements. Therefore, it is worthwhile to develop a framework to enhance the degree of standardization and integration between designer and manufacturer to obtain better cost and time trade-off and to implement the framework in practice.

This chapter first introduces the background of IBSs and its characteristics. After identifying the challenges in implementation of higher degree of prefabrication, the objectives of the research is presented. This chapter is continued by research methodology and thesis organization.

1.1. Background

Industrialization in the building industry has been pursued to increase construction productivity as well as the buildability and quality of buildings. The idea of using Industrialized Building System (IBS) in Singapore was first proposed during the early sixties because of market demand to build a large number of apartments in a short period of time using prefabrication. It is still being pursued even more actively in the face of increasing construction costs and the need to improve construction productivity due to limited resources and lack of skilled workers.

A study was carried out in Singapore comparing the economic benefits between the IBS and the conventional construction method in 1997. Its findings indicated that the use of IBS components provides labour savings of 46.5% as compared to conventional methods, thus reducing the dependency on foreign workers (Cheong, 1997). Further to this, the construction of "Bayshore Condominium" in Singapore indicated that the construction cycle time for each floor using conventional cast in-situ method was 22 days, which was 14 days more than using prefabrication based on similar site constraints and management experiences (Cheong, 1996).

In the perspective of construction, industrialization is part of a wider modernization process through the development of modern methods of design, production and technology system to mechanize construction operations. Industrialization is defined by

Warszawski (1999) as an investment in equipment, facilities, and technology with the purpose of increasing output, saving manual labor, and improving quality.

An Industrialized Building System (IBS) is defined as a system in which building components such as beam, column, wall, slab and stairs are produced in a factory or at a construction site under controlled environment to improve quality and to minimize on-site activities. Although IBS is almost known as prefabrication of building elements in a factory or at construction site, it is a combination of software and hardware elements to design, plan, produce, transport and erect all building components using industrialized processes.

Generally, the software elements focus on processes rather than products which include development of standardized components, production planning, establishment of manufacturing and assembly layout and process, allocation of resources and materials and definition of a building designer conceptual framework. In other words, the software elements provide essential requirements to expand industrialization.

Meanwhile, the hardware elements of IBS study the inter-relationships and connection of building elements. Hardware elements of a building system can be classified in different ways, depending on design, material, manufacturer and etc. For example, in precast concrete systems different types of precast elements may be employed. These systems can be classified according to the geometrical configuration of elements to: 1) Linear, 2) Planar, and 3) Three-dimensional or box. This classification is based on geometrical characteristics of the elements; however, it is not always precise or exhaustive. Different classification of the hardware elements has been reported based on construction material, design technology, production technology and construction techniques. These systems are discussed in the literature review in detail.

An extensive literature review on the area of IBS shows that the economy of prefabrication depends very much on the standardization and the integration between building design and the resources and the method employed by a particular precaster and contractor, who will eventually be selected to manufacture and erect the prefabricated components.

Various designs, multiple products, multi-stage operations, advanced production technology, fluctuating demand, and limited resources are a few characteristics of today's prefabrication industry which are not fully satisfied in the existing integration and standardization level. Standardization provides faster production, lower cost, and more efficient assembly of elements due to uniform dimensions that eliminate costly and time-consuming custom-made applications while still allowing multiple configurations.

In this regard, Tatum et al. (1987) proposed four basic levels of standardization where prefabrication can occur: total building prefabrication, system prefabrication, components prefabrication and elements prefabrication as depicted in Figure 1.1.

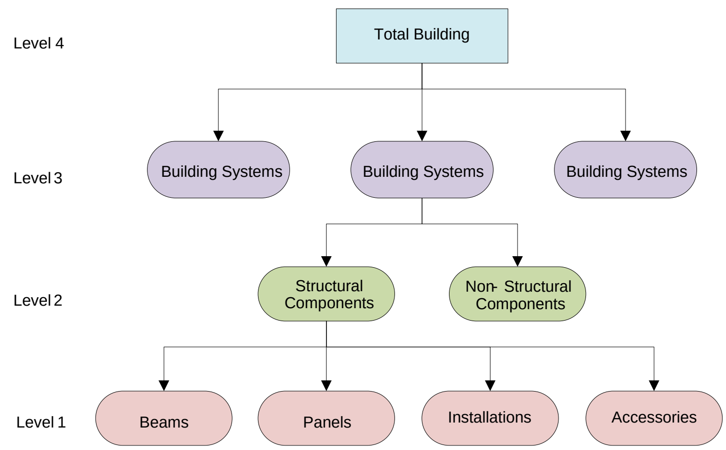


Figure 1.1: Possible Levels of Prefabrication (adopted from Tatum et al. (1987))

Figure 1.1 shows that the scope of standardization in prefabrication ranges from the production of individual elements of a building to the prefabrication of a complete building. If elemental prefabrication can be combined into bigger components, there would be less on-site activities, less handling and installation cost so that the total cost of production to installation could possibly be reduced. Furthermore, if prefabricated elements is not limited to particular shapes and sizes which is offered by producer, designers are flexible to consider functional requirements of spaces and aesthetic aspects of their design.

The highest level of prefabrication is modularization at level 4. From previous studies, the following advantages may be attributed to modularization: short build times, superior quality of joints and connections, economy of scale, environmentally less sensitive, safer construction, and reduced site labor requirement and professional fees (Montes et al., 2011; Polat, 2010; Rogan and Lawson, 2000). In this regard, Girmscheid

(2010) indicates that a greater value could be added to IBS by decreasing the degree of standardization and producing more complex components and modules due to rationalization effects in the production plant (Figure 1.2).

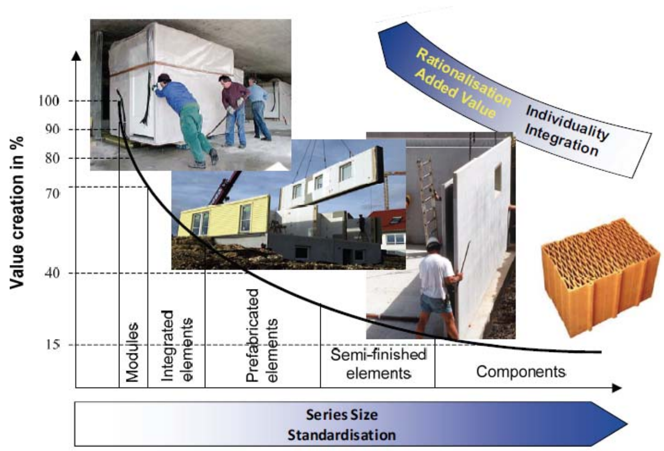


Figure 1.2: Impact of higher degree of prefabrication on project value creation (Source: Girmscheid (2010))

Literature reviewed shows that several attempts have been made to move beyond conventional individual approach to higher degree of prefabrication in IBS. Although these studies are not limited to any specific material such as concrete. For example, Friedman (1997) studied the potential for prefabrication systems to reduce cost of projects through combination of building elements including exterior walls, floors, partition, and structural walls to obtain a higher prefabricated wall system. Their analysis demonstrated that the proposed wall systems provide competitive alternatives to conventional prefabrication method.

There is, however, no comprehensive approach or model that has yet been reported in which higher level of prefabrication (component or modular level) has been evaluated numerically considering other key factors of industrialization such as standardization, design-production integration, flexibility of design, transportation and installation constraints for all types of structures. Therefore there is definite lack of analytical tools to evaluate the optimal degree of prefabrication in terms of standardization and resource consumption for project stakeholders to enhance the economy of IBS.

1.2. Challenges for Higher degree of Prefabrication in IBS

Although the AEC industry has been aware of the importance of industrialization in building systems for decades and even developed various decision support systems, framework and programs to improve economic benefits of the IBS in projects, the systematic approach to improve integration and standardization of prefabricated structure is still limited. A major reason is that the focus in the area of industrialization is mainly on direct methods of prefabrication production. AEC practitioners still encounter at least four challenges in successful implementation of industrialization in building industry.

Firstly, most of the AEC industry practitioners study IBS as a product, not as a process. However, literature review on definition and characteristics of IBS reveals that the scope of IBS is not restricted to the final product. Researchers and practitioners indicate that the IBS is formed by several processes that lead to a final product. This approach to industrialization requires focusing only on improvement of prefabrication and its planning instead of the processes to improve degree of prefabrication.

Secondly, moving beyond the production of individual building elements towards configuring a higher level of prefabrication needs a long-term planning programme in which all participants of the project are involved. The high initial cost of complex mould for setting up the manufacturing plant as well as the cost of transportation will reduce the margin of profit in short-term planning.

Thirdly, employing a high level of prefabrication may reduce the flexibility of designer due to the existing pre-defined coordination between designer and producer. However if a systematic and flexible coordination exists between designer, producer and contractor, a trade-off between flexibility and standardization can be achieved so that project time and cost can be reduced while the flexibility is maintained.

Finally, such a standardization and integration between designer and producer requires a custom-make building product data model. Basically, the necessary information for the product data model includes a number of attributes about a building and its components, which determine both spatial and functional behaviors of interest to various AEC disciplines. The spatial information describes the building components' geometry and topology (i.e. dimensions, locations, and relations among the components), while the functional data represents all the other discipline-specific properties of the components (e.g. structural, design constraints, and building regulations). Evidently, a successful building product data model should be capable of not only facilitating data exchange, but also of enabling automation of design and construction processes, e.g. automated generation of feasible configuration of precast element considering design, production, transportation, and lifting constraints. Although existing data models are capable of handling spatial and semantic information and data exchange, these models should be enriched and customized to handle the functional analysis required in the context of this study. For example, IFC is one of the most well-known and common data models which is capable of describing spatial and semantic information of building elements. However, the topological relationships among building elements (that is required for forming higher level components) in a building is not considered in IFC.

1.3. Research Hypothesis

Moving beyond individual building elements toward modularization may result in increased value added to the project, while employing a higher degree of prefabrication may reduce flexibility in design, transportation and assembly due to component size and shape. In addition, having more complex and unique components may reduce standardization and mass production.

The main objective of standardization is to use resources efficiently. With higher degree of prefabrication, this objective still can be achieved through substitution of complex moulds to produce various smaller building elements (component groups). Grouping components to form more complex components depends on geometric/topological relationships of building elements, their physical (size, shape, material) and functional (structural, non-structural) properties. The process of grouping cannot be implemented manually. To this end, an intelligent framework with advanced computational tools and algorithms is required to achieve a higher degree of prefabrication. This framework enhances coordination between designer and producer considering physical, functional and spatial properties of building elements. Furthermore, this framework must give maximum flexibility to designer while considering standardization of products, and constructability criteria.

This research aims to develop the necessary concepts, framework and tools to improve design-production integration and standardization to facilitate a higher degree of prefabrication while achieving an optimized resource consumption and production plan. In the other words, the objective of this research is to develop a framework to configure the grouping of elements into higher level components in order to minimize the total number of components so as to reduce transportation and lifting costs and at the same time to maximize the number of identical components for reason of economy of scale in terms of mould fabrication cost.

The research will develop a framework which is called Component Configuration of Prefabricated Structures (CoCoPS) and the prototype CoCoPS will be implemented in a case study to assess the research findings. The overall process in CoCoPS framework is presented schematically in Figure 1.3. Its main purpose is to obtain an optimal degree of prefabrication for each particular project in order to minimize project cost and time.

To this end, the framework needs to extract general, semantic, and geometrical properties of building elements from an intelligent parametric model. Topological relationships among building elements are deduced so that all possible configurations of building elements are generated. A rule-based database is utilized to check constructability of the generated configuration in terms of design, production, transportation, and installation. Changing the degree of prefabrication may affect all project stages. Thus, two optimization models are proposed and implemented on the set of feasible configurations. The first model optimizes the resource consumption in prefab plant. The second model optimizes the transportation resources of feasible configurations.

To achieve this goal, this research project is intended more specifically for delivering the following objectives:

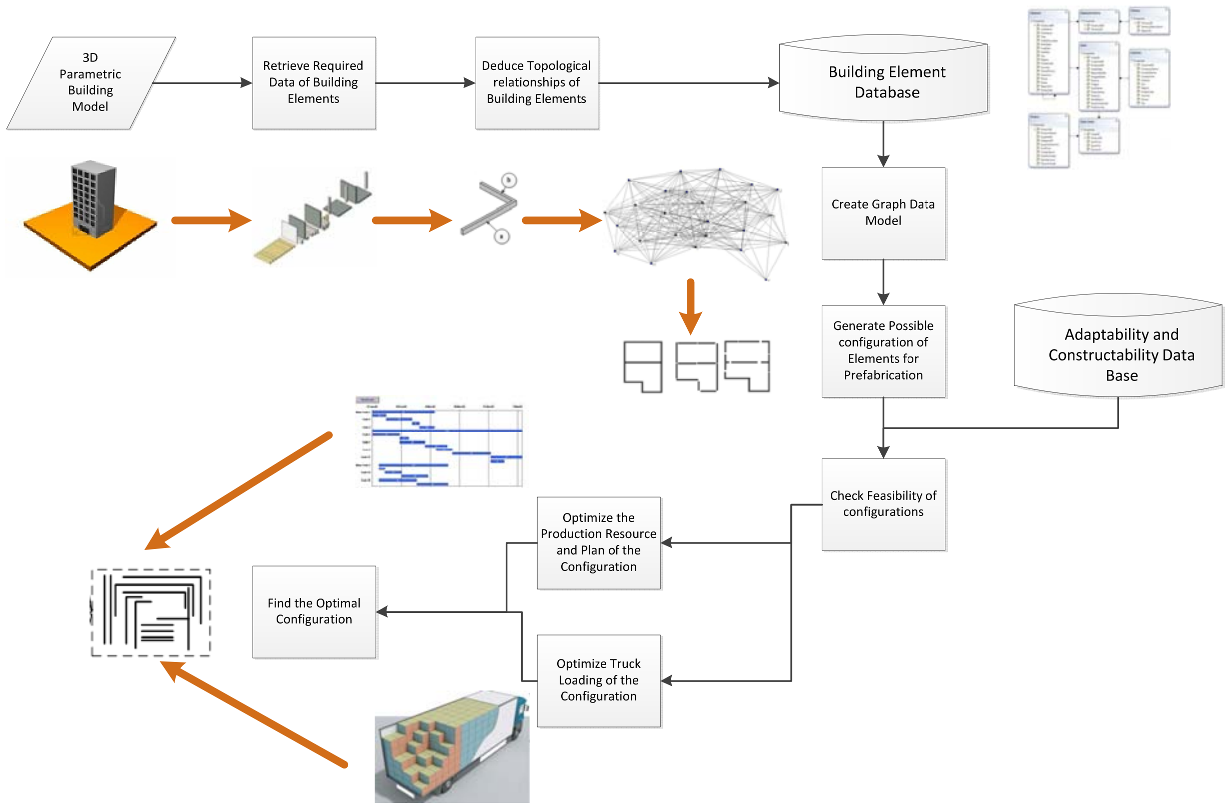


Figure 1.3: The proposed CoCoPS framework

1- Develop Component Configuration of Prefabricated Structures (CoCoPS) framework.

The present research attempts to develop an intelligent framework which extracts the required data including physical and geometric properties from an existing designed building through IFC. After mapping these data in a new Graph Data Model (GDM), the CoCoPS utilizes advanced graph algorithms and AI tools to generate a set of possible prefabrication configurations. The generated configurations are compared against the knowledge embedded constructability rules to obtain all feasible configurations due to practical constraints for production, transportation and assembly. An evaluation system is employed to achieve the optimal configuration in terms of resource utilization.

2- Build a Graph Data Model (GDM) to represent topological relationships and semantic information of building elements in a micro-spatial environments.

Literature review has indicated that there are several existing models to represent geometry and topological relationships among building elements through topological primitives. However, these models are inadequate to handle various queries and complex network analysis. A novel Graph Data Model is proposed in this research to abstract and represent 3D objects and their relationships in a graph in which nodes denote 3D objects; while edges represent the topological relations. The GDM is enhanced by adding semantic information as weights to nodes and edges to be able to handle wide ranges of queries. Unlike the conventional topological data models, the GDM is constructed based on a new approach of deduction and representation of topological relationships. In this approach, the presentation of 3D objects does not start with geometrical information. Instead, only the topological information is used to denote the engineering objects of a building in the first step. A complete expression of building elements and spaces is set up where no geometrical information needs to be specified. In the second step, geometrical information is considered only for further queries and analysis. This approach may effectively maintain topological consistency within micro-spatial environment.

3- Design an automatic algorithm for deduction of topological relationships among building elements using IFC.

Topological relationships of building elements should be deduced to assign to edges in GDM. Formal representation of 3D objects in general and topological data in particular of building elements is a complex and challenging task due to the fact that representation of topologies and dimensionalities to express spatial information of building elements varies with the expected applications. Furthermore, different design tasks require different types of topological information. In order to automatically generate the GDM, an algorithm is needed to extract such topological relationships among building elements. This study develops an IFC-based deduction algorithm for the four major types of topological information as follows:

- Connectivity: One object connects to the other
- Separation: Objects are separate from each other
- Containment: One object lies within the other
- Intersection: One object intersects the other

4- Propose a precast production planning optimization model using concepts of prefabrication configuration and component groups.

The proposed framework generates complex building components which require complex casting mould in prefabrication plants. Mould fabrication is the most expensive resource in prefabrication plants which have certain capacity and life cycle. In this regard, an optimization model is required to maximize mould usage in its life cycle to produce a range of standard building components. This research attempts to develop a mathematical model to adopt two new concepts which are prefabrication configuration and component groups to optimize precast production resource cost and provide an optimal production plan considering construction site demands.

5- Put up a new customized approach for loading irregular 3D objects into trucks to optimize transportation of prefabricated components.

The transportation of irregular three-dimensional components which are produced with a higher degree of prefabrication can be extremely costly. This research seeks to extend the general Bin Packing Problems (BPP) and Container Loading Problems (CLP) optimization models which are available for 2D and 3D cuboid objects, for placement of 3D irregular-shaped components into the truck. An emerging approach to this problem is investigated in which heuristic sliding method is used for placement of objects into a container through a voxel representation according to the sequence generated by genetic algorithm.

6- To assess the developed framework and proposed concepts and methodologies through a case study.

This research will validate the developed framework, concepts and algorithms with a case study. This case study is the construction of a residential complex comprising 91 blocks of 10- storeys with different floor plans.

1.4. Research Methodology

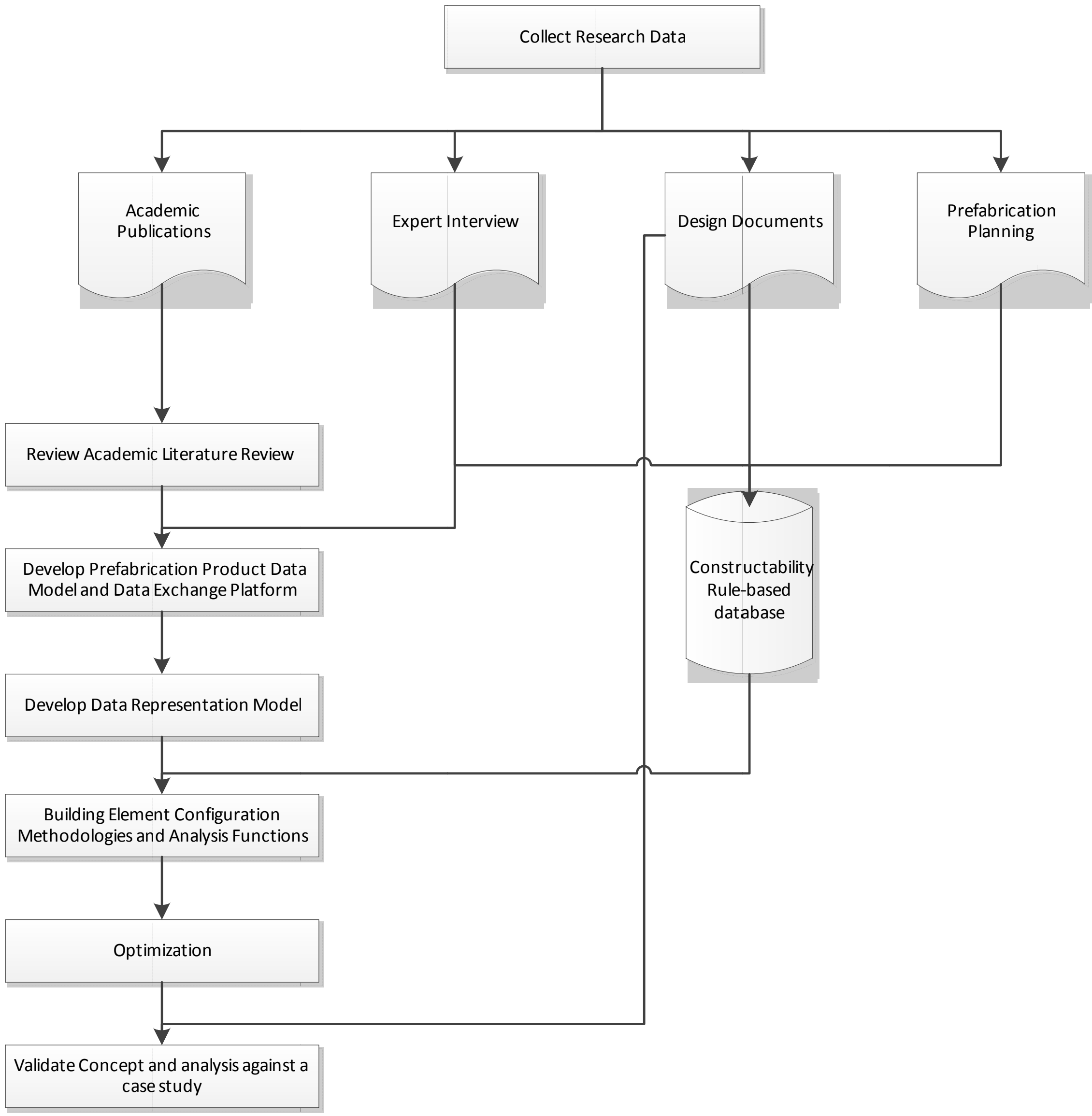


Figure 1.4: Research Methodology

The research methodology of this study is shown in Figure 1.4. In the initial stage, different types of research data related to Industrialized Building System (IBS) were collected for the subsequent study. The collected data can be classified into four categories: Academic publications, expert interviews, design drawings, and prefabrication planning. There were more than 200 papers and book chapters collected and reviewed of which two-thirds of them have been referenced by the present study.

The author visited consultant companies, prefabrication plants and attended coordination meetings between clients, designers and precasters to understand the current practice of industrialization from design to construction as well as to collect real data and design drawings. Personal interviews were arranged individually with planners, designers and precasters and construction experts over two study travel to actual construction site in Iran. We looked for local manufacturers to select a pilot in Singapore. Since all manufacturers were producing components for several projects, it was not possible to change their set up for the proposed model. However, the selected pilot site in Iran has dedicated prefab plant so that it is possible and feasible to run the test on it. These interviews and visits were required to understand the current gaps in industrialization process and to identify potentials for improvement in prefabrication. Moreover, the collected data together with the international regulations and standards for transportation and lifting of prefabricated elements were utilized to establish the constructability database in this research. This database is used to select feasible configuration of elements among all generated configurations.

Having presented the contributions of this research, the author could motivate the client to apply the research results in the prefabrication planning of “Diamond Town” which is one of the biggest residential complexes in Iran.

1.4.1. Academic Literature Review

The review of academic literatures covers such research fields as IBS characteristics, prefabrication, product data models, building information modeling, data exchange platforms mainly IFC, topological representation, prefabrication planning, planning optimization, and container loading problem. This literature review provides a solid background for the present study to explore the lack of developed models and approaches for standardization and design-production integration in prefabrication industry.

1.4.2. Component Configuration for Prefabricated Structures (CoCoPS)

A continuous information flow with a neutral three dimensional building information model (BIM) is required for successful planning for higher degree of prefabrication. This framework requires a product data model to describe semantic and physical characteristics of prefabricated building elements. The basic elements of the developed framework includes: data structure, data abstraction, representation schema, data exchange platform, analysis functions, evaluation criteria and validation.

1.4.3. Develop data representation model

The Graph Data Model (GDM) was developed before executing the CoCoPS analysis functions since the GDM can abstract and represent 3D objects and their relationships in an effective and efficient manner. The developed GDM contains a data management system that enables performing wide range of queries and network analysis through its graph data structure.

1.4.4. Building element configuration: Methodologies and Functions

The concept, methodology, and analysis functions required to configure building elements to obtain a higher degree of prefabrication is explored in this part of the research. These functions are presented in the constructability analyzer module in the proposed framework including Graph Generator, Comparator and Standardization. Since the number of building elements in a medium-sized building is more than a hundred, millions of configuration of elements can be generated. Heuristic and artificial intelligent methods such as tabu search, and sub-graph isomorphism were utilized to obtain feasible and practical configuration of building elements in this study.

1.4.5. Optimization

After generating a set of feasible configurations, an evaluation system is required to find the best configuration. To this end the most important and costly steps of prefabrication which are production and transportation were identified through literature review and expert interviews. Therefore two optimization models were developed to optimize the resource required for implementing each feasible configuration. The first optimization model aims to minimize the production resource cost and to find out the optimal production plan. The second optimization model was developed to minimize the number of trucks required to transport produced components to the construction site. The second optimization model is important because the generated components in each configuration may have irregular shapes so that the placement into truck must be optimized to reduce transportation cost. The robustness of the proposed optimization models are proved by real-project case studies.

1.4.6. Evaluate Concept, Framework and Optimization models using a Case Study

The developed concept, framework and optimization model were validated using the design drawings and data collected from industry experts. The results were compared to the traditional individual approach in prefabrication to demonstrate that a higher level of prefabrication can improve the industrialization process.

1.5. Organization of Thesis

This research will first present a literature review on industrialized building systems, prefabrication, standardization of building components, design-production coordination, and integrated planning for prefabrication coordination in Chapter 2. Chapter 3 will present the developed framework to move beyond individual approach in prefabrication towards a higher degree of component configurations. In Chapter 3, the analysis functions and methodologies required to configure building elements to obtain a higher level of prefabrication will be discussed. Chapter 4 will discuss how the graph data model for representing building elements and their relationships are established. It will also describe the proposed Graph Data Model (GDM) and the algorithm for automatic deduction of topological relationships among building elements using IFC. Chapter 5 and Chapter 6 provide the evaluation basis for generated configurations in developed framework. Chapter 5 focuses on the model developed for optimization of resource in prefabrication plant. In Chapter 6, a novel optimization approach is proposed for extension of the general Container Loading Problem (CLP) to place produced three-dimensional irregular-shaped components into trucks. Chapter 7 will present a case study to demonstrate how the concept, framework, and functions are utilized to improve design-production process of an actual project. Finally, Chapter 8 will conclude the thesis and provide suggestions for further research and development.

CHAPTER 2: LITERATURE REVIEW

The present study relates to such research fields as Industrialized Building System (IBS), off-site fabrication, standardization of building components in IBS, design-production coordination, and integrated framework for IBS coordination. The review aims to introduce the relations of IBS, prefabrication and standardization to assess the strengths and weaknesses of current methodologies for standardization in prefabrication and IBS. These research findings indicate that the previous studies in the context of standardization are qualitative and these models are inadequate to satisfy the current needs of prefabrication industry. The improvement of prefabrication industry using higher degree of standardization is rarely considered. It is noted that subsequent chapters will also present reviews of literature relevant to the content of the chapters.

2.1. Industrialized Building System (IBS)

Researchers in the field of construction often state that industrialization has many advantages in comparison with conventional construction methods. Currently, there is a wide definition on the term industrialization. For example, industrialization is defined in its general format by Warszawski (1999) as an investment in equipment, facilities, and technology with the purpose of increasing output, saving manual labour, and improving quality. However, in the perspective of construction, industrialization is part of a wider modernization process through the development of modern methods of design, production and technology system to mechanize construction operations. Industrialization focuses on mass production and mainly factory production (Lessing, 2006). The

International Council for Research and Innovation in Building and Construction (CIB) defined industrialized construction as a generic process of standardization and rationalization of the work processes in the industry to reach cost efficiency, higher productivity and quality (CIB, 2010). A more elegant definition for industrialized construction is a change of thinking and practices to improve the construction to produce a high quality, customized built environment, through an integrated process, optimizing standardization, organization, cost value, mechanization and automation (CIB, 2010).

One of the efforts towards industrialization in construction is through the introduction of Industrialized Building System (IBS). In this regard, the term building system is defined by Warszawski (1999) as a set of interconnected elements that join together to enable the designated performance of building. It is also characterized as a set of interrelated elements that act together to enable designated performance of building. In a wider sense, it may include several managerial, technological and operational procedures for the design, production and installation of these elements for this purpose (Sarja, 1998b).

So far, there has been no uniform definition of IBS. However, a few definitions by researchers who studied in this field were found through literature. The term IBS is poorly defined, often interchangeably with other terms such as ‘offsite’ and ‘prefabrication’. Their precise definitions depend heavily on the user’s experience and understanding, which vary from user to user.

The lack of a uniform definition and uncertainty in context and boundary of IBS contribute to its misunderstanding. Thus, a workable definition needs to be developed. The earliest definition of IBS found in literature is by Dietz (1946) that defined IBS as a total integration of all subsystems and components into an overall process fully utilizing industrialized production, transportation and assembly methods. This definition was improved by adding a structured planning and standardization (1990). The term system in IBS includes balance combination between software and hardware elements. The software elements include system design, which is complex process of studying the requirement of the end user, the design constraints, the development of standardize building component and transportation and construction limitations (2011). However, the hardware elements of IBS study the inter-relationships and connection of building elements (1986).

The benefits derived from the implementation of IBS are the speeding up of the construction process, the integration of sustainability strategies, optimum use of materials, repetitive and reuse of molds, less waste of materials, reduction of wastages during construction, and the minimization of hazards and risks (Hassim et al., 2009; Kamar et al., 2009; Nawi et al., 2009; Thanoon et al., 2003).

In an attempt to understand the poor diffusion of IBS, some researchers have investigated and identified a number of barriers to the effective implementation of the IBS. Relevant literature sources (Kamar et. al., 2009; Hamid et. al., 2008; Thanoon et. al., 2003; Hussein, 2007; CIMP, 2007; Nawi et. al., 2007; Chung, 2006; and CIDB, 2003) reveal the main generic barriers as current market attitude for mass customization, constructability issue, manufacturing capability issues, cost and financial issues. To address this gap, the current IBS methods and characteristics should be improved using quantitative methods.

Generally, there are several categories of definition and characteristics for IBS. Among the collected academic literatures, about sixteen authors have defined IBS as both a process and a product (Bock, 2004; Sarja, 1998b; Warszawski, 1999). However, only five authors defined IBS as either a product or a process. Apart from the differences in definitions, it can be observed that seven characteristics are mentioned as the main processes and products of IBS which consists: prefabrication, standardization, industrialized production, transportation and assembly techniques, design-production coordination, process integration and structured planning.

Among these, standardization and prefabrication have been addressed as foundation of industrialization that may result in higher quality, lower construction time and possible lower cost in projects (CIRIA, 2000; Gibb, 2001b). It should be mentioned that the prefabrication is considered as process (design, planning, production, transportation and installation of prefabricated elements) and product (building components). Gibb (2001) and CIRIA (2000) indicate that the improvement of IBS is tied to degree of standardization,

Researchers in this area highlights that most influential methods for improvement of IBS are prefabrication and standardization. Therefore several studies have been done to define these concepts, and to identify their benefits and barriers clearly (Gibb, 2001a; Gibb and Isack, 2003; Gibb, 1999a; Girmscheid, 2010; Goodier and Gibb, 2004; Jaillon and Poon, 2009). These studies utilized surveys and case studies to highlight the practical drivers and issues in implementation of IBS in practice. Moreover, they have provided codes and standards for design, production and installation to improve implementation of IBS (Warszawski 1999). Although some of the aspects of IBS such as design and production of prefabricated components have been studies in depth to improve IBS, there is no integrated framework in which processes of IBS are analyzed and optimized quantitatively in relation to each other. Most of the processes such as standardization and process integration are evaluated qualitatively.

An integrated framework is required to use new building data models, and IT tools to analyze processes and products during the construction life cycle to assess the IBS implementation and possibilities of improvement in current construction industry.

It is mentioned that IBS is certainly not new to the construction industry and new research area. However, current influential construction trends, such as the increasing interest use of information technology and BIM have caused many researchers to reconsider their studies. In fact, these technologies may facilitate the implementation of an integrated framework and may recommend IBS as an opportunity for breakthrough achievement.

2.2. Prefabrication

2.2.1. Definition

Prefabrication (as a general name) is seen as one of the methods of improving construction in the past decades over the world (Egan, 1998). The CIRIA (Construction Industry Research and Information Association) acknowledges that construction must be re-engineered and so that a much greater emphasis on off-site assembly is one of the key topics for improving the performance in construction industry.

Prefabrication is defined as a manufacturing process generally taking place at a specialized facility, in which various materials are joined to form a component part of final installation (Tatum et al, 1986). Prefabrication started a few hundred years ago and improved in the middle of the last century. A full discussion of the historical development of prefabrication is out of the scope of this study, but has been published in (Gibb, 2001a; Groák, 1992; Herbert, 1984; Sarja, 1998a; Tatum et al., 1987; Warszawski, 1999).

The theoretical context of prefabrication also has been covered extensively in literature (BSRIA, 1999; CIRIA, 1997; Gibb, 1999b; Warszawski, 1999). In the last few years there has been an increase in interest and study with much work from organizations such as (ASCE, 2011; CIRIA, 1999, 2000). For example, ASCE studied the interests, drivers, and barriers of prefabrication through a survey from more than 80 companies and case studies. The Housing Forum also published a series of reports in which a number of necessary actions for prefabrication stakeholders has been introduced (Housing Forum, 2002).

2.2.2. Prefabrication Classification

Prefabrication has been classified by researchers as shown in Table 2.1. Gibb (1999b) proposed three categories of off-site fabrication such as, non-volumetric, volumetric and modular building; although he argued that the line dividing each type is flexible. This flexibility opens a new approach in which the type of prefabrication can be determined and optimized according to project design, production and installation facilities. The given examples and case studies in Gibb (1999) also prove this idea. For example, in the Scottish Widow project volumetric prefabrication was used; non-volumetric elements also attached to the produced elements due to special structural design. Spatial configuration of prefabricated elements, interface consistency, connections and on-site fabrication are the issues that have been considered by Gibb (1999b) in his classification.

Tatum et al. (1987) identified different levels of prefabrication within a building, for example, total building, building system, building components and building elements (Figure 1.1). Figure 1.1 shows that the scope of building elements in prefabrication ranges from the production of individual elements of a building to the prefabrication of a complete building. If elemental prefabrication can be combined into bigger components, there would be less handling so that the total cost of production to installation could possibly be reduced. The highest level of prefabrication is modularization at level 4.

From previous studies, the following advantages may be attributed to modularization: short build times, superior quality, economy of scale, environment friendly, use on infill sites, safer construction, and reduced site labor requirement, and possible cost reduction (Montes et al., 2011; Polat, 2010; Rogan and Lawson, 2000).

Thanoon et al. (2003), studied Industrial Building Systems (IBS) in Malaysia. They indicated that modularization is the most efficient building system in terms of standardization, tolerances, mass production and coordination.

In this regard, Girmscheid (2010) and Gibb (1999b) also studied the impacts of standardization in various degrees of prefabrication. Their research indicates that the value added to a project is significantly increases if prefabrication is standardized from elemental level to module level (modularization) (Figure 2.1).

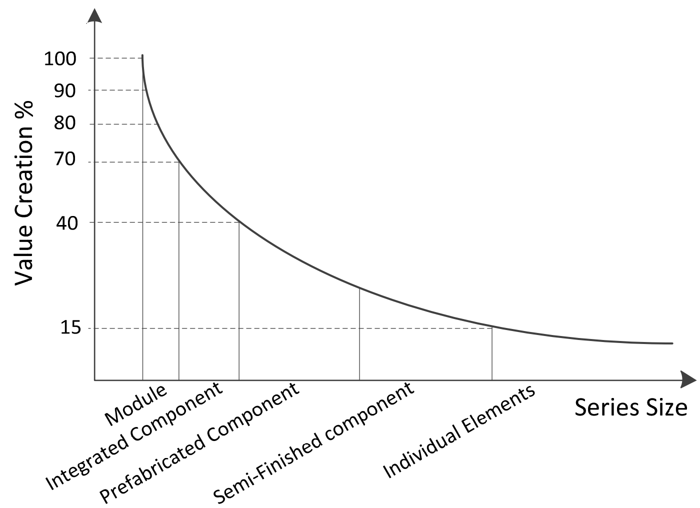


Figure 2.1: Value added to the project (percentile) according to degree of prefabrication adopted from (Girmscheid, 2010)

Although it has been mentioned that different degrees of prefabrication may lead to certain added values to the project; an effort is required to identify and evaluate these values. Therefore a decision making and analysis tool is required to obtain the desired degree of prefabrication. The benefits and barriers of each level of prefabrication for a particular project must be considered in this decision making process.

Table 2.1: Classification of prefabrication systems

Author	*Date*	*Classification System*	*Description*
Gibb	1999	Non-Volumetric	Items that do not enclose usable space structural frame or cladding internal partitions, pans of building services, distribution ductwork or pipe work
		Volumetric	Units that enclose usable space, but do not of themselves constitute the whole building. These units are used for facilities, such as toilets/washrooms, plant rooms, building services risers and lifts, and are installed in new or existing buildings. The units generally do not provide any support for the building structure.
		Modular Building	Units that form a complete building or part of a building, and includes the structure and the building envelope.
Tatum et al.	1986	1st Level Building Elements	The smallest pieces of a building that can be discussed in terms of prefabrication. They include windows, doors and panels. It is now standard practice to use prefabricated elements in almost any type of construction projects.
		2nd Level Building Components	Parts comprised in a system. Components may be combined into a large number of configurations to form systems.
		3rd Level Building Systems	A system within a total building, such as the building structure or the building envelope. The total building is formed by joining several systems together
		4th Level Total Building	Essential features of the building, such as structure, building enclosure and roof.
Warsawski	1999	Linear System	Main structural elements: columns, beams, frames, or trusses made of plain or pre-stressed concrete. Their important feature is the capacity to transfer heavy loads over large span. Application: warehouses, industrial buildings, sport facilities, etc.
		Planar System	They are frequently used and are employing planar and panel-shaped elements for floor slabs, vertical supports, partitions, and exterior walls. They fulfill interior and exterior space enclosure functions. They may include finish work such as exterior finish, thermal insulation, electrical conduits and fixtures, plumbing, door and window frames, etc. Application: residential buildings, offices, schools, hotels, and similar buildings with moderate loads and large amount of finish works.

2.2.3. Benefits and Barriers:

The descriptive pros and cons of prefabrication can be found in previous studies (Warzawski, 2000; White, 1965; Russell, 1981; Herbert, 1984; Tatum et al., 1986; Groa´k, 1992; Gann, 1996; Gibb, 1999). According to a comprehensive literature review on prefabrication and surveys from practitioners, the drivers of using prefabrication among project stakeholders have been identified and categorized. Among these, time, quality and cost have been identified as the most referred benefits (Figure 2.2). For example in Figure 2.2, 38 references addressed "Time Reduction" as a benefit of prefabrication. Other benefits reported include productivity, less congested construction site, simplified construction process and environmental friendship.

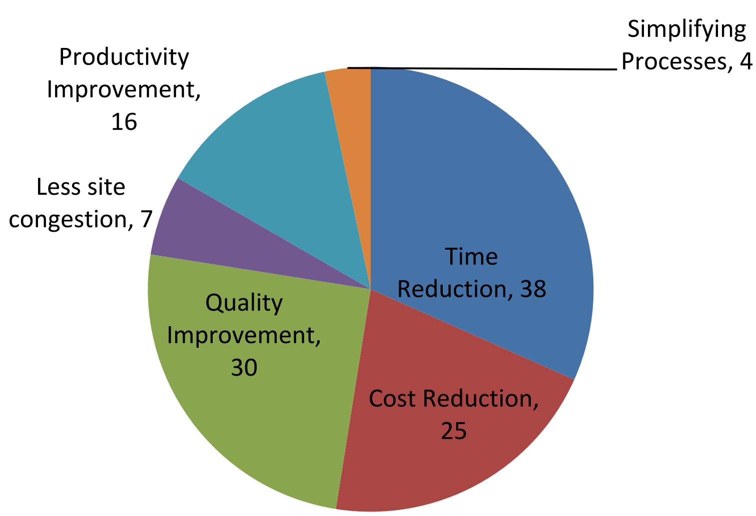


Figure 2.2: Benefits of Prefabrication reported from researchers and practitioners

The first benefit is identified as project time. Time is an issue for almost all clients. Time may incur significant costs if clients/contractors failed to meet project completion dates. It may also bring early income from shorter completion date. There are other hidden benefits in early project completion date such as risk reduction or market changing.

Quality is identified as the second most cited benefit of using prefabrication. It is obvious that factory-made products have higher quality than the on-site building components. Moreover, prefabricated building components have better consistency to fit together correctly. The factory made products can be visited before installation and it will reduce the risk of repetition work and fixing cost.

From literature review, cost has been identified as the third important benefit of prefabrication (Gibb 2001, CIRIA 2000). Cost of project is always the main concern of project participants. Groak (1992) and Gibb (2001) have found from surveys that in prefabrication total cost is more important than specific elemental production cost. For example, although it might be more expensive to prefabricate a building component; other savings such as reduced preliminary costs, installation cost, the absence of scaffolding, transportation cost, lower site congestion, and controlled-environment should be included in cost comparisons and may compensate for the additional 'bottom-line' cost.

The cost of prefabrication is usually studied as production cost (Chan and Hu, 2002a; Dawood, 1995) and the costs that imposed from other stages of prefabrication are neglected. Planning for prefabrication is also performed on production plant to improve resource utilization and production time (Hao, 2007; Huang et al., 2005; Leu and Hwang, 2002). For example, Warszawski (2000) studied various models for estimation and optimization of prefabrication cost. He collected a wide range of models that use GA, constraint programming, and linear programming for prefabrication plants in his book developed by Dawood (1995), Hao (2007), Chan and Hu (2002b), Zhai et al. (2008). In these models, economic impacts, financial improvements, factory layout improvements, and planning optimization for manufacturer have been discussed. However, the cost of transportation and installation for different categories and methods of prefabrication has been rarely addressed.

Cost of transportation and storage is studied by Kuo-Chuan Shih (2005). They considered the behavioral pattern of the storage and transportation of the prefabrication factory, and then constructed an optimized pattern for cost and planning of prefabrication. Although they considered two other stages in their model, they still optimized the project from factory perspective (Huang et al., 2005).

It can be concluded that the added value to the project can be studied from quality, time and cost perspectives. It is obvious that the quality of product will be increased if the prefabrication is implemented in modularization level rather than the elemental level because a bigger portion of the structure is fabricated off-site and in the factory environment (Figure 2.3a). Total construction time is also reduced if bigger components is produced off-site and assembled on construction site (Al-Hussein et al., 2005; Warszawski and Ishai, 1982) (Figure 2.3b).

The cost of different degrees of prefabrication is not easy to predict. In order to find out the impacts of prefabrication on project cost an integrated model is required to consider all cost constituents and project stakeholders. Moreover, employing a higher degree of prefabrication may result in less flexibility in design, transportation and assembly due to more potential constraints. The cost of prefabrication in different phases of the project and the flexibility of design can be studied and optimized using standardization (Gibb and Isack, 2003; Gibb, 2001b). If the inter-relation of standardization and prefabrication can be modeled, an integrated model for the optimal degree of prefabrication for each particular project can be identified.

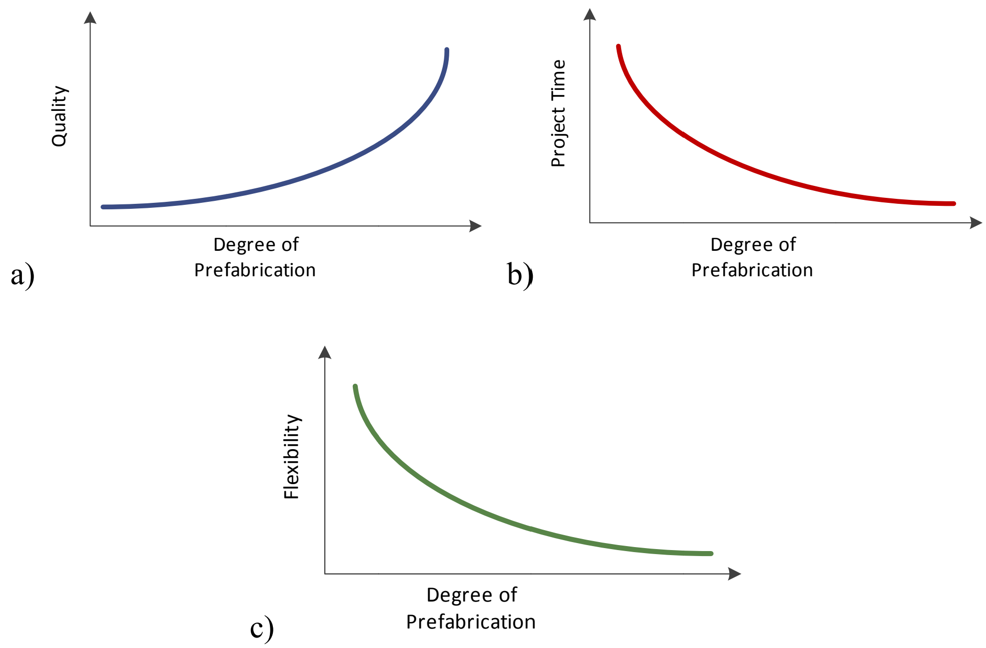


Figure 2.3: a) Quality of construction vs. degree of prefabrication, b) Total project completion vs degree of prefabrication, c) Flexibility of design vs degree of prefabrication

2.3. Standardization

Various researchers from previous studies have presented standardization as a design philosophy (e.g. White, 1965; Russell, 1981; Gropius and Wachsmann, 1984). Standardization is defined by CIRIA as "the extensive use of components, methods or processes in which there is regularity, repetition and a background of successful practice" (CIRIA, 1997). Production resources can be used in the most efficient manner if the output is standardized. Then the production process, machinery, and worker's training can be focused on the particular characteristics of the product. CIRIA indicates that project participants must attempt to standardize processes and products across the construction cycle rather than production (CIRIA 1997).

Standardization of processes and procedures enabled project teams to streamline the overall construction process, which was claimed to reduce cost and required resources. This potential exists even on individual projects, but clearly could be even greater for larger clients with repeat orders (Gibb, 2001b).

Understanding and commitment to standardization and prefabrication by all parties was considered vital. Design decisions generally had to be made earlier than for conventional construction, and critical information had to be established at the earliest possible stage. In an effort to address historical criticisms of standardization and prefabrication, project teams worked hard to ensure that standardization and prefabrication increased design choice, facilitated controlled innovation and ensured work of quality, aesthetic appeal and distinction.

The idea of standardization is back to standard bricks and tiles. However, the application of standardization in prefabrication is back to the development of the factory-based prefabrication of building components (Groák, 1992, p.134).

The use of modular frameworks has existed for centuries (e.g. in Europe since the early Renaissance). However, the real motivation to combine standardization with systematic building grew with the development of off-site fabrication shops and factory-based building component industry (Groák, 1992, p.134).

The CIRIA survey found that the greatest benefit is obtained if standardization and prefabrication are used together (CIRIA, 2000). The CIRIA research included a number of contemporary case studies. CIRIA was aimed to demonstrate real applications of standardization rather than theoretical advantages.

A comprehensive literature review on history of standardization and prefabrication reveals that there are two fundamental drivers for standardization in prefabrication, namely: market demand (an urgent response to market demand) and cost reduction.

2.3.1. Market Demand

The response to the market demand started with exporting prefabricated house from industrial revolution when European nations expanded. This driver of standardization requires mass production. For example, in the mid-19th century Brunel developed standardized, prefabricated hospitals for the war in the Crimea. In Singapore, there has been a recent urgent need for housing that has been addressed by standardization and prefabrication (Gibb, 1999). However, today's market demand requires more flexibility in architectural design and floor plan rather than need for house (CIRIA, 1999). The new market demand requires the maximum flexibility.

There is always a challenge between architectures and other project stakeholders about flexibility of design (Groák, 1992; Herbert, 1984). Historically, those studied on standardization (e.g. Herbert, 1984, p.48; Gibb 1999, Warszawski 2000) have always struggled to resolve the conflict between uniformity and variation, between maximum standardization and flexibility. This conflict has still not been resolved.

This has led to the demand for mass customization rather than mass production. This is where the benefits of mass production can creatively be combined with systems that offer greater choice for the individual customer; provide improved control of the total construction process, and flexibility of assembly options. Therefore the traditional perspective of standardization that concentrates on mass production using the available library of products is inadequate to respond to clients' demands.

2.3.2. Project Cost

The cost reduction is always addressed as the main concern of clients. Gibb and Isack (2010) made a survey on benefits and barriers of prefabrication and standardization on more than 80 construction stakeholders. They reported that cost is the third important driving force of prefabrication after time and quality, especially for repeat-order clients. But many respondents noted that total cost is more important than specific elemental product cost.

Gibb (1999) indicates that the most important area for standardization is the construction life cycle (from design to construction) rather than the components themselves. Their report highlights that the cost of adopting standardization and prefabrication on a project is not limited to the production cost. However, most of the research studies only focus on the effects of standardization on mass production.

To achieve the maximum reduction in cost and maximum flexibility to satisfy both project participants and customers, a new attitude is required in which manufacturers must trade-off the need to achieve economies of scale in the production of standardized building components with economies of scope in various stages of design and assembly in order to provide flexibility to satisfy consumer needs.

It can be found from literature review that standardization is not limited to maximization of number of component repetition (mass production) but it can be optimization (trade-off between mass production and mass customization). For example, Groák (1992, p.34) showed that for one design, there is only one optimum production method. This does not properly recognize the extraordinary variety of production units – and their flexible combinations in the building industry. He mentioned that the maximum standardization is not the only solution to every situation. Clearly there is a balance between maximum off-site fabrication and the additional costs of transportation (especially where some of the units are largely empty) (Bottom et al., 1994; Gibb, 1999a; Herbert, 1984) also indicate that the ideal is one of optimization for standardization in prefabrication industry rather than maximization.

An optimal level of standardization in prefabrication is achieved when designers are flexible to satisfy market demands, prefabricated elements are standardized to allow the producer to rely on the economies of scale, and total number of components is reduced so that the transportation and assembly cost will be decreased. To this end, an enhanced coordination between designer and producer should be defined in which, flexibility of designer, standardization of products, and constructability criteria are considered.

Although researchers in the area of architecture, manufacturing and construction attempted to introduce degree of standardization as a key solution for cost reduction and demand satisfaction in prefabrication industry; these explanation are not sufficient to practically identify the degree of standardization in projects. The lack of quantitative method in analysis of prefabrication and standardization in a project leads to higher cost and resource consumption in projects.

For instance, Gibb (1999b) indicates that the higher degree of standardization leads to less project production, transportation and installation cost. As an example, Gibb (1999b) shows the degree of standardization for a precast cladding units in which, based on timber moulds, the minimum repetition for cost-effective manufacture is around ten units and the optimum repetition is 30 units. It is obvious that after this the added benefit gained from more repetition becomes less significant. However, having less standardization in the project will cause more production cost (Figure 2.4).

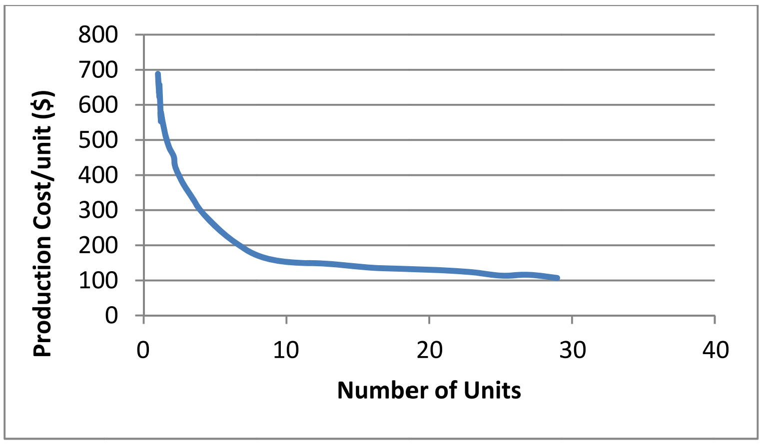


Figure 2.4: cost of production per number of repetition

Warszawski (1999) also indicated that the larger amount of repetition in prefabricated component results in the smaller production cost per element. He concluded that even better economic results can be obtained if the prefabricated components can be utilized for other projects. In other words, if the degree of standardization increases from make-to-order to make-to-stock the economy of scale will be further increased.

2.4. Integrated Frameworks between Standardization and Prefabrication

This section attempts to review the existing method and models to integrate prefabrication and standardization in order to evaluate different degrees of prefabrication.

The emergence of new IT tools may enhance the performance of industrialization by employing a higher degree of prefabrication through integration of standardization and design-production coordination. Several scholars indicate that the regular industrialization of buildings can be further augmented by development of new methods and IT systems and integration between standardization and prefabrication (Helena Johnsson, 2007; Linus Malmgren, 2011; Retik and Warszawski, 1994).

Retik and Warszawski (1994) presented a knowledge-based automated system to enable the architects, even with a poor knowledge on precast building systems, to pursue his design goals within manufacturer's constraints . The proposed system by Retik and Warszawski (1994) brought a degree of flexibility to designer to explore and assess various design options. However, their system could handle 2D drawings in which all building elements have the same elevations in each floor. Moreover, this system only deals with orthogonal buildings in which building elements follow two main perpendicular directions. However the proposed framework in current study is able to handle 3D models with no constraint on floor plan.

Europe established a mega research projects being undertaken in the prefabrication domain, with automation as the core which is called FutureHome project. The FutureHome project started in 1998 and was completed in 2002. The most important element of the project is the development of a system which is called "AUTMOD3". This system is an automatic modular construction software environment (Diez et al., 2007b). AUTOMOD3 attempts to integrate architectural design, planning and simulation tools through a commercial CAD program. The developed system covers a wide range of modules including structural modules, roof modules, and façade modules in 2D. Diez, Padrón et al. propose an automatic modularization decision system which determines the type and number of modules to be employed in a project. AUTOMOD seeks to minimize the number of modules considering the module dimension limitations due to prefabrication, crane capacity, and transportation limitations (Diez et al., 2007a). Diez et al. (2007a) utilized a module library in their integrated construction automation model to provide a degree of flexibility for modularized buildings. They developed a 2D CAD-based solution to modularize buildings according to module library and facilitate the design-production coordination. Their system automatically detects defined zones within drawings and uses available 3D modules in the library to configure group of components surrounded by each zone.

There are two methods of modular design in their developed system. The first method is based on traditional architectural design in which plans are drawn in a CAD system. These drawings are processed using a modular division procedure to obtain the modules needed for the building. If there is any problem during the modular division due to the complexity of the design, the system requires the architect to make appropriate corrections. In the second method, the design is performed using a library of modules. All information is stored in a central database, which allows its access from other processes in the system.

Although Diez's system automatically configures components to make modules, it uses limited functional modules such as kitchen and bathroom which may not be utilized in non-residential buildings. Their proposed framework is based on 2D drawings and it processes drawings in the plan so that the third dimensions and its constraints such as height of elements are not mentioned in this project. Further to this, the modular dimension, vertical dimension and other dimensional coordination standards are not considered in this system.

2.5. Contribution to this study

Previous literature review enables us to draw some conclusions and consequently identify several issues that this research attempts to address:

Firstly, IBS has been reported as a potential to improve construction industry. Standardization and prefabrication have been addressed as foundation of industrialization that may result in higher quality, lower construction time and possible lower cost in projects. The proposed methods for improvement of IBS are descriptive and vulnerable. There is no integrated framework in which processes and products of IBS are analyzed and optimized quantitatively in relation to each other. It is necessary to extend the research in standardization and prefabrication to use new IT tools and building information models to quantify these processes efficiently. With this regard, the present research exploits the advantages of building product data models, IFC, graph theory and optimization models to establish a quantitative evaluation framework for standardization and prefabrication in IBS.

Secondly, researchers in prefabrication area highlight that different degree of prefabrication may increase added values to the project. An effort is required to identify and evaluate these values. A decision making and analysis tool is required to obtain the desired degree of prefabrication. The benefits and barriers of each level of prefabrication for a particular project must be considered in this decision making process. This study, develops a Graph Data Model (GDM) by which all possible configuration of building elements are obtained. GDM together with a rule-based constructability database and a library of preferred shapes are used to identify benefits and barriers of various level of prefabrication for a particular project.

Thirdly, it has been identified that Time, Quality, and Cost are the most addressed benefits to the prefabrication. Project time will be reduced if a higher degree of prefabrication is employed. Quality of product will be increased by adopting modular components rather than prefabricated elements. According to the survey, project stakeholders are more concerned about the total project cost rather than specific elemental product cost. There are several stages (such as production, transportation, installation) and various cost factors (investment cost, direct and indirect cost, etc). Thus it is not easy to discuss about the cost impacts for various degrees of prefabrication and here is the area of research where optimization can be beneficially employed to improve prefabrication and IBS.

Fourthly, the cost of prefabrication is usually studied as production cost and the costs constituents that imposed from other stages of prefabrication are neglected. However, the total cost of construction cycle is required in order to evaluate prefabrication method. The cost of prefabrication is determined by standardization of products. Researchers on standardization have always struggled to resolve the conflict between uniformity and variation, between maximum standardization and flexibility. This conflict has still not been resolved. This research is aimed to re-engineer the standardization on the light of current advanced computational tools and building data models. A two-stage optimization model is proposed in this study to evaluate the cost of standardization in various configuration of building elements for prefabrication. In this model, production, transportation and installation of building component are considered.

CHAPTER 3: FRAMEWORK FOR COMPONENT CONFIGURATION OF PREFABRICATED STRUCTURE (CoCoPS)

This chapter presents framework of the Component Configuration of Prefabricated Structures (CoCoPS) framework. CoCoPS aims to generate all feasible prefabrication configuration of building elements and tries to find the optimal configuration in terms of resource utilization. Essentially, CoCoPS is designed to extract topological relationships and geometrical properties of building elements from an IFC file and map this data to a proposed topological Graph Data Model (GDM). Using graph algorithms such as Depth First search (DFS) and graph isomorphism, all possible configurations are generated and compared against production and construction rules. A simple optimization model is incorporated to verify the importance of configuration in precast production and installation using an illustrative example.

3.1. Structure of CoCoPS framework

The present research develops a framework to employ a higher degree of prefabrication in order to balance between flexibility in design and resource utilization in production, transportation and construction assembly. Since the framework aims to automatically generate all feasible configuration of building elements and find the optimal configuration, a continuous flow of information is required. A continuous information flow requires a neutral three dimensional building information model (BIM). Although the proposed framework can be adopted for any parametric-based modelling platform, Industry Foundation Class (IFC) has been selected in this study because it is one of the most notable and widely accepted product data model in building industry. It represents a digital and parametric information structure of the objects making up a building, capturing the form, behavior, and relationships of the components and assemblies within the building. It allows efficient sharing and exchange project information. The overall design of the proposed framework is shown in Figure 3.1 which comprises five key features as depicted in the shaded elements of the model. The following are the key features:

(1) An internal data structure developed from extracted physical and spatial properties from an IFC model, such as dimensions, materials and topological relations.

(2) A graph data representation model for geometrical and semantic information mapping.

(3) Generation of prefabrication configurations using graph search and isomorphism algorithms.

(4) Comparison of these configurations against existing production and constructability criteria to obtain feasible set of configurations.

(5) Optimization of resources required to produce, transport and install feasible configurations such as mould, transportation and lifting.

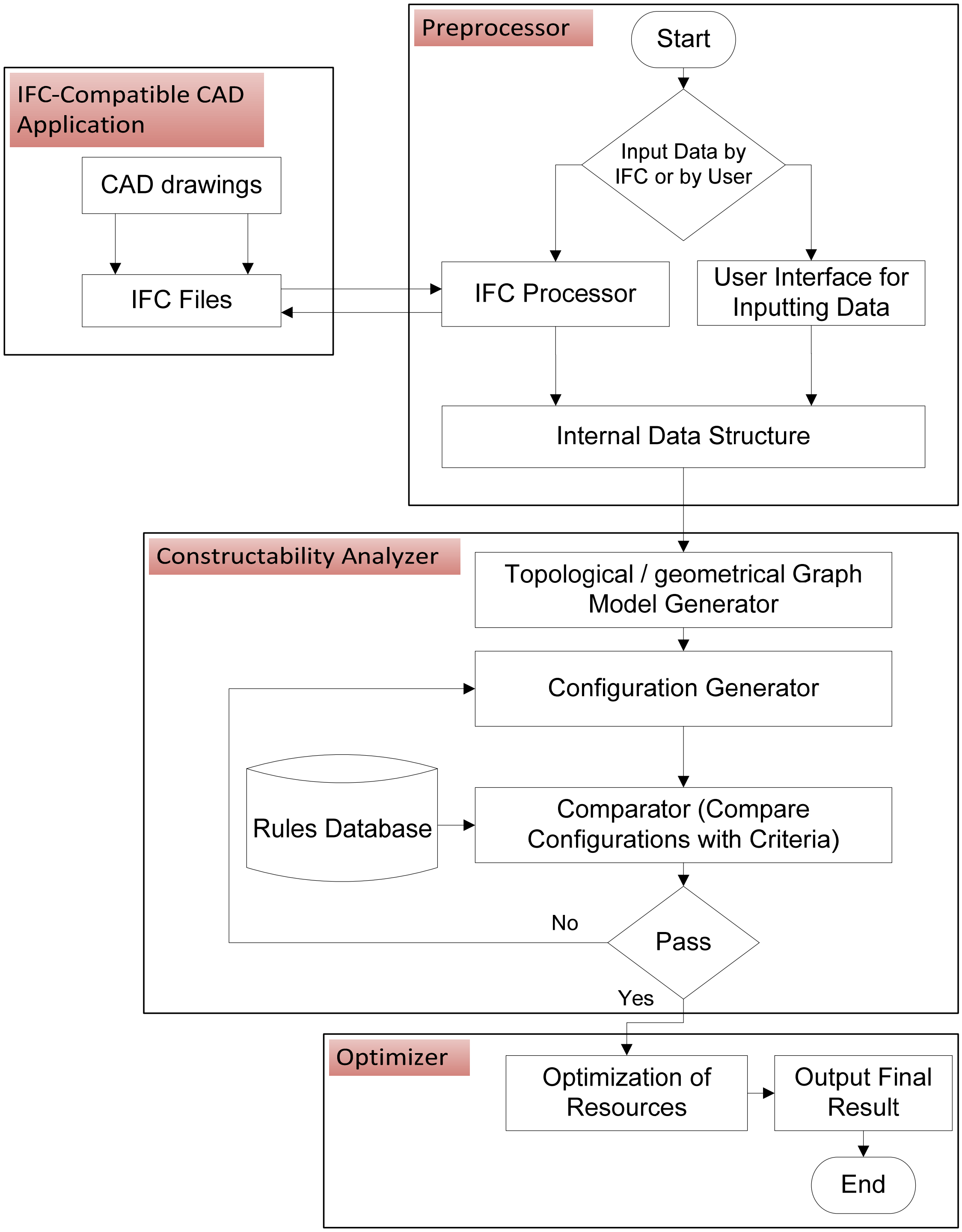


Figure 3.1: Proposed System Architecture for CoCoPS

As shown in Figure 3.1, the framework comprises four modules elaborated as follows:

3.1.1. An IFC-compatible CAD application

This is a CAD tool which can export 3D CAD drawings into IFC files. It can also read IFC files and transfers them into 3D CAD drawings. Autodesk's Revit with its IFC2x utility and ArchiCAD with its add-on interface are two typical IFC-compatible CAD applications available in the market.

3.1.2. Preprocessor

Essentially, the preprocessor is a functional unit in CoCoPS that parses the input data acquired from an IFC file and then builds an internal data structure (middle layer) to provide easy data access for its receptor, the graph model generator. The preprocessor consists of:

- An IFC processor, which imports building elements data, including topology/geometry properties, dimensions, materials, and functionality, from an IFC file into the framework. Furthermore, it can modify the IFC files, for example, creating, retrieving, and deleting IFC instances, and setting or editing attribute values for IFC instances.
- A user interface which is a graphical interface for the user to input/modify data from IFC model or manually. It is designed to maintain the database, especially to provide missing information from IFC files.

- An internal data structure, which is used to map the data from the IFC processor and the user interface for inputting data into the memory. The data structure is a relational database comprising two tables, one describing the relationships between elements and the other consisting of geometry, dimensions and material properties of the building elements which is termed as Semantic Data Table (SDT). SDT includes: ID, Type of element, Dimension, Material, Representation, Weight, and user defined data.

3.1.3. The Constructability Analyzer

The Constructability Analyzer is the main module of the proposed CoCoPS comprising three key functions namely, Graph Model Generator, Configuration Generator, and Comparator. These functions are described as follows.

- The first function is the "Graph Model Generator" which transposes the geometry and topological relationships of building elements from the database to a graph model. This function uses the internal data structure together with the IFC file to automatically deduce the topological relationships among building elements. Building elements, their relationships and SDT are mapped into a novel data representation model which is called Graph Data Model (GDM). Development of GDM is described in Chapter 4. The output of this module is a node to node adjacency matrix.

- The second function is the “Configuration Generator”. It is the main processor of the framework which traverses the graph model to generate various configurations of prefabrication. Configuration generator function combines individual building elements to obtain higher level components using graph search and isomorphism algorithms. This process mainly relies on the topological relationships of building elements and their physical and functional properties. Configuration Generator receives adjacency matrix and generates a binary configuration string (S)
- The third function is the “Comparator” which compares the generated configurations with constructability criteria from the rule database and components library to obtain a set of feasible configurations. The rule database contains dimension coordination, vertical dimension and the criteria governing manufacturer’s capability to produce different shapes and sizes of prefabricated components. Further to these, constructability rules comprise information about transportation limitations, lifting constraints, safety issues, environmental limitations, and site restrictions. The component library comprises a collection of preferred feasible components. This library is created by designers based on their past experience, local standards, and manufacturer capabilities. The library of preferred shapes can be employed in the comparator function to reduce the domain of feasible configurations to a set of preferred components. For example Figure 3.2 shows three feasible components according to the structural and transportation criteria. However the U-Shape and T-Shape components are easier for manufacturer to produce than the irregular one.

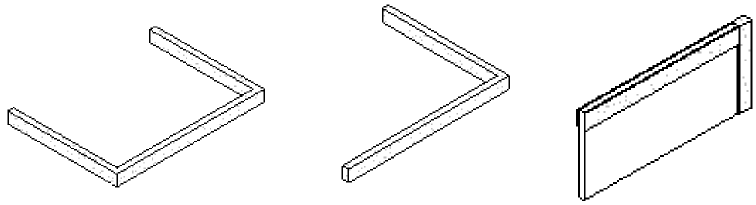


Figure 3.2: Sample feasible component from library of preferred shapes

3.1.4. Optimizer

It is a functional module in the framework that processes resource utilization of the generated configurations. This module determines the optimum configuration among set of feasible configurations in terms of minimum mould, transportation and installation cost. The Optimizer module comprises:

- Production resource optimization model which is a mathematical model to determine the number of moulds of each type that is required for each feasible configuration. Since the higher degree of prefabrication requires complex casting moulds, a novel approach should be developed to maximize mould utilization to produce complex and simple prefabricated components. The proposed model is a Mix-Integer Programming problem. This model is solved using CPLEX solver implemented on General Algebraic Modeling System (GAMS).
- Transportation optimization model which is utilized to optimize arrangement of components on truck in order to minimize the number of trucks required for each feasible configuration. Truck loading optimization is important due to the complex and irregular shape of generated building components. Trucks are used to move components from the factory to the construction site. The issue of transportation is modeled as a 3D container loading problem for irregular objects.

A novel approach is developed to use voxels for modeling components. A new sliding algorithm together with Genetic Algorithm is adopted to solve this model.

- Installation resource optimization model which is a mathematical model to represent contractor's constraint for installation. The constraints are installation sequence, lifting capacity, lifting points and on-site storage. This model uses local and global standards and empirical data for modeling. The model is solved using CPLEX solver implemented on General Algebraic Modeling System (GAMS).

The Optimizer module integrates the abovementioned models at a higher-level to obtain the optimal configuration of building elements and to provide and an integrated planning system. The optimization models are described in chapters 5 and 6 in detail.

3.2. Proposed framework development

This section presents the development of the two main functional modules of CoCoPS framework. The constructability analyzer and optimizer are the two functional modules of the proposed framework. The first functional module generates a set of feasible configurations; and the second evaluates the cost of generated configurations. The flow of information is provided by the underlying IFC data exchange platform in the framework. The following describes the development of the functional modules. The constructability analyzer is developed using three concepts namely, geometry/topological graph representation models, graph isomorphism algorithm and tabu search. The optimizer is based on integer programming and heuristic and AI methods.

3.2.1. Graph Model Generator

As can be seen in Figure 3.1, the graph model generator is one of three main functions (the others being configuration generator and comparator function) in Constructability Analyzer module in the proposed framework. It utilized a graph geometry/topological model to map the topological and geometric data of the building elements into a graph so that graph theory may be applied for query and further analysis. The development of GDM, its structure and the algorithms used for deduction of topological relationships among building elements are discussed in detail in Chapter 4.

3.2.1.1. Topological representation using graph theory-GDM

The topological model is closely related to the representation of spatial relationships among objects in a building. Over the last fifteen years, topological models for n-dimensional objects have been developed by a number of researchers (Pigot and Hazelton, 1992; Rikkers, 1993). The topological models have their advantages in avoiding redundant storage, maintaining data consistency and performing specific topological operations such as overlap, intersections, etc (Raper, 2000). A number of topological systems such as Formal Data Structure (FDS) and TEtrahedral Network (TEN) have been developed to implement 3D objects based on boundary representations (B-rep) (Raper, 2000). However these models are inefficient for maintaining topological consistency among building elements because of complex geometric computations that involve geometric properties of local neighborhood topology. To overcome this problem, combinatorial data models such as graph models have been developed (Lee and Kwan, 2005). Instead of representing the topological relationships between topological primitives as in the aforementioned topological models, the graph models present the topological relationships among 3D objects by drawing a dual graph interpreting the adjacency-connectivity relation between these 3D objects.

The Graph Data Model (GDM) proposed in this study defines the spatial relationships of the building elements within an entire building in a global graph, G. The nodes or vertices, V, in the graph represent the building elements, and the edges, E, represent the connectivity between the elements. A set of sub-graphs (H_i) defines the possible configurations for prefabrication. Each graph therefore comprises two sets of entities, V and E, defining the vertices and edges, respectively (Equation 3.1).

$$G = (V(G), E(G))$$

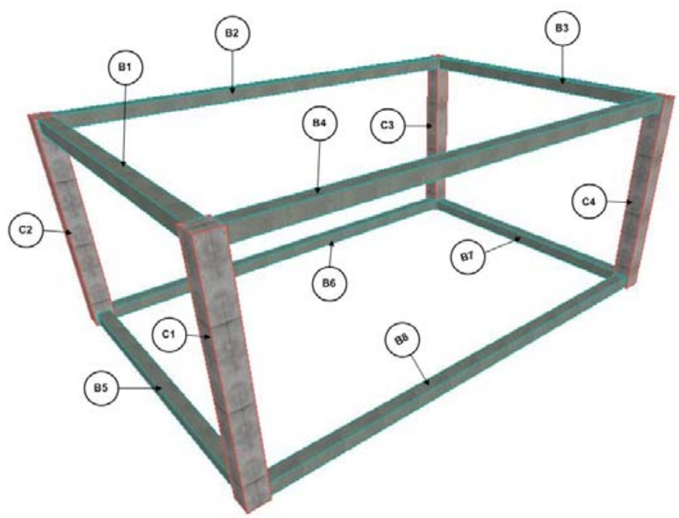
$$H_i = (V(H_i), E(H_i)) \tag{3.1}$$

The graph H_i is a sub-graph of global graph G since $V(H_i) \subseteq V(G)$ and $E(H_i) \subseteq E(G)$. Figure 3.3a shows an example of a simple concrete frame comprising 8 beams (4 beams at the top and 4 beams at the bottom) and 4 columns connecting both top and bottom beam frames, which is translated to a graph using 12 nodes and 24 edges as depicted in Figure 3.3b.

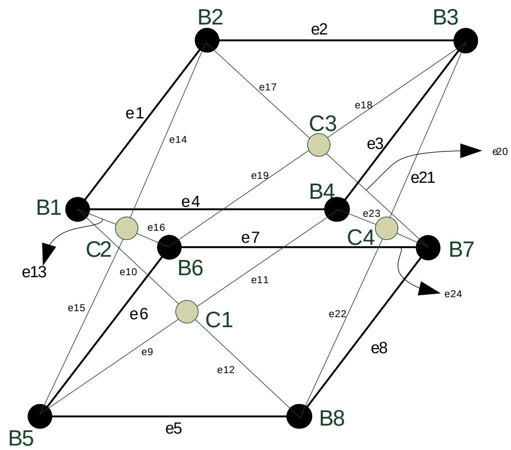
There are several methods for graph representation such as Node-Node Adjacency, Node-Arc adjacency, or Adjacency List. The most common representation of graphs for computational purpose is Node to Node Adjacency matrix. The main alternative to the adjacency matrix is the adjacency list. Because each entry in the adjacency matrix requires only one bit, it can be represented in a very compact way. Operations with a graph represented by an adjacency matrix are faster. But if a graph is large we can't use such big matrix to represent a graph, so we should use collection of adjacency lists, which is more compact. Using adjacency lists is preferable, when a graph is sparse, i.e. $|E|$ is much less than $|V|^2$, but if $|E|$ is close to $|V|^2$, or number of edges cannot be forecasted it is better to choose adjacency matrix, because in any case we should use $O(|V|^2)$ memory.

In this study, for computational processing and storage, the graphs are stored as an adjacency matrix. With N building elements, the Adjacency Matrix is an NxN matrix Adj. The value in Adj [i, j] denotes whether element j is an immediate successor of element i or not. For undirected and unweighted graphs, Adj [i, j] can be true or false, (1 or 0) as follows:

$$Adj[i,j] = \begin{cases} 1 \text{ if } i \text{ is directly connected to } j \\ 0 \text{ otherwise} \end{cases} \tag{3.2}$$



(a)



(b)

Figure 3.3 : (a) Sample concrete structure (2 beam grids and 4 columns) (b) Relationship of beams and columns using graph model

The building elements in the IFC file forms the set of nodes in GDM. As a logical data model, GDM is a pure graph representing the adjacency and connectivity relationships among the internal elements of a building. In order to implement network-based analysis such as graph traversal algorithms in the GDM for constructability analysis by comparing with constructability rules, the logical network model needs to be complemented by a 3D Geometric Network Model (GNM) that accurately represents the geometry properties such as placement, shape, length and weight. These properties are tied to the corresponding nodes and edges of the GDM.

3.2.1.2. Mapping of Geometrical properties into the graph model-GNM

A major problem of handling geometrical and topological information is the interpretation and utilization of these data with respect to specific tasks such as Depth First Search (DFS) or graph isomorphism for constructability control. The IFC data format comprises both geometrical and topological information necessary to build the structured component graphs. The required geometric information includes the shape, size, placement and orientation of the building elements.

The definitions of geometric representations in the IFC Releases 2.0 and 2.x are quite close to the well approved STEP geometric definition of ISO 10303- 42 (1994). Any object in IFC with a geometric representation has two attributes: *ObjectPlacement* and *Representation*.

ObjectPlacement stores the placement information of an object. The default method is the relative placement, given by two entities *IfcLocalPlacement* and *IfcAxis2placement3D*. *IfcLocalPlacement* defines relationships of coordinate systems recursively from higher level to lower level, while *IfcAxis2placement3D* supplies the position and orientation of the corresponding coordinate systems and at the lowest level depicts the position and orientation of the object in the local coordinate system. Figure 3.4 shows the definition of the placement of an object (a wall) in IFC.

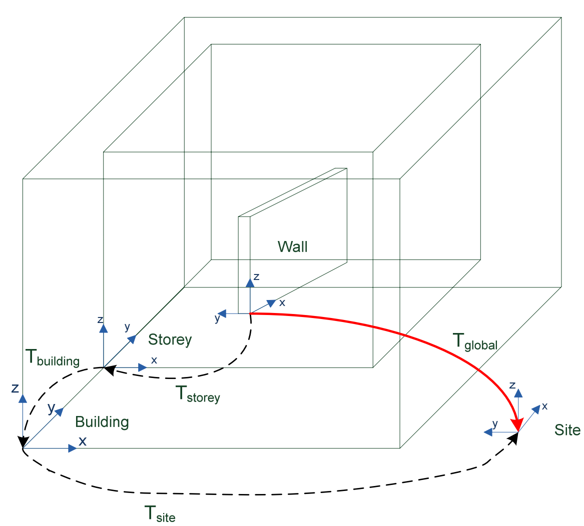


Figure 3.4: The concept of *IfcLocalPlacement* and its relations to other local coordinate systems

In GDM, the local placement of an object is transferred to the global placement for the deduction of topological relationship. Thus the main step in using geometrical properties of building elements is the transformation of coordinates from local to the global. Local placement is represented by an affine transformation (T), which consists of a linear transformation (the rotation of the local coordinates followed by a translation). The algorithm for converting local coordinates to the global one is briefly described in following paragraphs. The algorithm will be discussed in details in Chapter 4.

Using ordinary vector algebra, it is possible to transform the local coordinates (T_{Storey}, $T_{building}$, T_{site}) to the global coordinates in a single step by multiplying all local transformations (rotations and translations) to obtain a single global transformation T_{Global} (Equation 3.3).

$$T_{Global} = T_{Wall} \times T_{Storey} \times T_{Building} \times T_{Site} \tag{3.3}$$

Representation essentially describes the shape of building elements in IFC. The shape is important in the GNM where it is used for constructability analysis. The explicit geometric representation is defined by coordination of the object. Geometric representation can also be attribute-driven. Generally, IFC uses attribute-driven geometric definitions for objects. Figure 3.5 shows the representation of a wall in its local coordinate. The attributes of the wall in Figure 3.5 are:

- Cross section profile: for this case, it is a rectangle with a height of YDim and a width of XDim
- Extrusion Direction: the direction in which the cross section is to be swept;
- Depth: the distance the cross section is to be swept;

The local coordinates of resultant shape, defined by its attributes, is then transformed into the global coordinate system and incorporated into GNM (Figure 3.5).

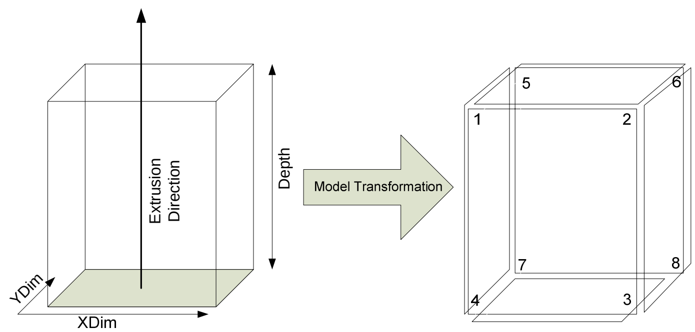


Figure 3.5: The definition of 3D geometry of an object and transformation to Boundary representation

Having placement and representation of 3D objects from IFC, the topological identifiers required for deducting the topological relationships among building elements can be calculated. The topological relationships of building elements are deduced automatically using the algorithm proposed by the author and will be described in detail in Chapter 4. Topological relationships form the set of edges of the GDM. Geometrical information and the other required properties such as volume, material are calculated and stored as a table in GNM for further analysis.

3.2.2. Configuration Generator

The Configuration Generator creates all possible combinations of components from the GDM. The set of edges of the GDM that depicts connection of building elements is transformed to a configuration string (S). Indices of the configuration string are IDs of edges and the value of the string is binary where "1" represents the existence of the edge and "0" otherwise (Figure 3.6a). The initial state of S is $S[i]=\overline{0}$ representing the first configuration in which all the building elements are prefabricated individually. Values of S are changed incrementally on a binary basis to generate all possible configurations. Each state of S, S_i, implies a new configuration graph which is a subset of the GDM. For example in Figure 3.6b, the sample structure (see Figure 3.3a) is divided into two sub-graphs (building components) given by S_i= {1,0,1,0,1,1,1,1,1,0,1,1,1,0,0,0} in which {e2,e4,e10,e14,e15,e16} are zero (disconnected) and the rest of edges are one (connected). This state implies that the main structure is fabricated using two smaller components, one of which comprises building elements {B1, B2, C2} and the other the remaining elements. The complete set of sub-graphs G={G_i} may be detected for each S_i using the Depth First Search (DFS) algorithm to traverse the connected elements in the configuration graph.

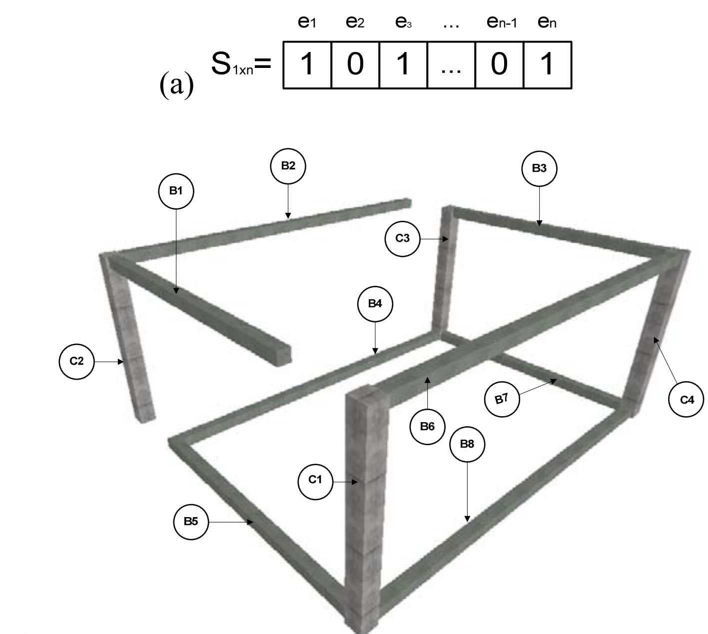


Figure 3.6: (a) Transformation of GDM to configuration string S, (b) Generated configuration of sample structure

The length of (S) increases by the number of building elements within a building. In order to enhance the computational efficiency of modelling configuration string and generating configurations, binary string is utilized with “False” and “True” representing “0” and “1” respectively.

3.2.3. Comparator

The last function in the Constructability Analyzer module tries to find a set of feasible configurations among all possible generated configurations using constructability rules, preferred component library and Tabu Search (TS) method. Thereafter, graph isomorphism algorithm is applied on each feasible configuration to classify similar components for optimizing mould fabrication, transportation and lifting cost.

Since the cardinality of possible solutions exponentially increases by the number of vertices in the GDM, a tabu search method is incorporated to reduce computational time.

A comprehensive literature review on tabu search can be found in Glover and Laguna (1993). Tabu search incorporates three general components: (i) short-term and long-term memory structures, (ii) tabu restrictions and aspiration criteria, and (iii) intensification and diversification strategies.

The tabu search algorithm developed herein based on this design is as follows:

Begin

Step 1. Find an initial feasible solution, normally S_i=**0**.

Step 2. Set Iterations=0, MaxIterations= 2^|n|; InfCounter=0; MaxInfCounter.

Step 3. While Iterations < MaxIterations then

(a) Choose an edge to be moved (connected or disconnected)

(b) Evaluate the current solution.

(c) If (New_Solution is not Tabu) AND (next move is feasible) then

Add solution to Feasible Solution list.

(d) If (New_Solution is not Tabu) AND (next move is infeasible) then

If InfCounter < MaxInfCounter then

Add New_Solution to tabu list and set *Tabu_Time(e)*

InfCounter : = InfCounter + 1; Iterations := Iterations + 1; repeat Step 3(a)

Else

InfCounter = 0; Step 3(e).

Else

Add Solution to tabu list and set *Tabu_Time(e)*; InfCounter: = 0.

(e) Iterations := Iterations + 1.

End

For the GDM setting, the tabu search procedure begins with an initial feasible starting solution (Step 1). The simplest initial solution would be prefabrication of individual building elements (S_i=**0**). If this configuration cannot satisfy the rules, the prefabrication of the whole project is infeasible.

Parameters of the tabu search algorithm including the maximum number of iterations (MaxIterations) and the infeasibility counter (InfCounter) are set in Step 2. The MaxIterations represents the maximum number of configurations that can be generated by configuration string (S). The infeasibility counter (InfCounter) allows traversing an infeasible problem space in consecutive iterations, in order to form tabu list.

The movement for changing the current solution to another solution (Step 3(a)) is based on the Graph Generator procedure discussed earlier. In the evaluation function as can be seen in Step 3(b), each sub-graph of current solution is checked against production and constructability criteria. If any sub-graph of a configuration S_i fails the criteria, the configuration is not feasible and the corresponding move will be assumed as a tabu. The criteria are rule-based as depicted in Figure 3.7 and include:

- Production: dimension, shape and weight requirements for mould fabrication, stock yard.
- Construction: dimension and weight requirements for transportation and erection of components (e.g. truck size, crane capacity and site accessibility)
- Structural and non-structural joints requirements for maintaining structural unity and avoiding in-situ concreting.

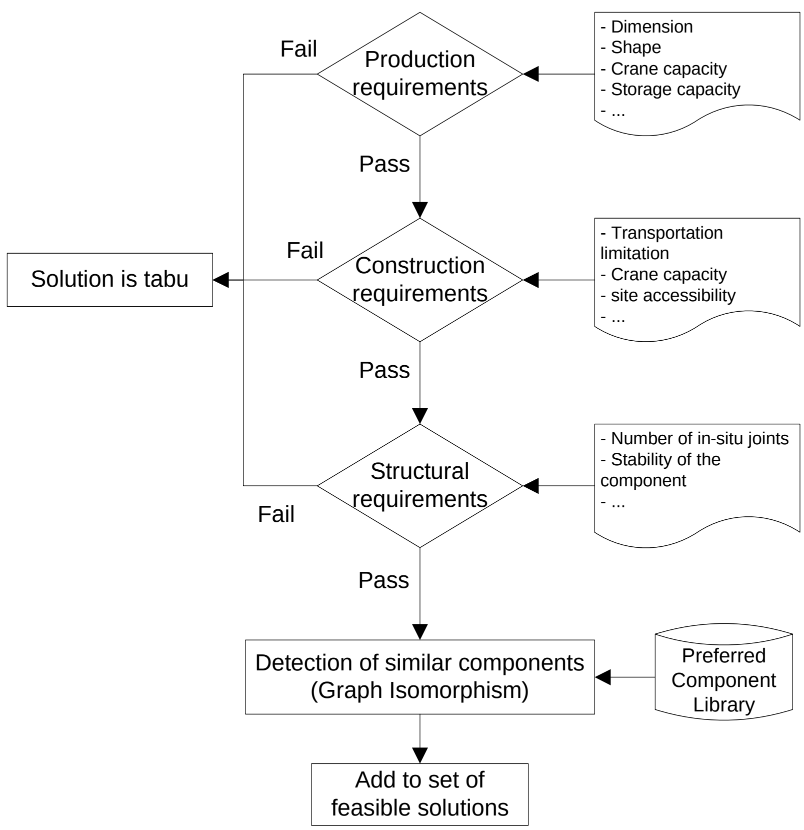


Figure 3.7: Process flowchart of evaluation function

Solutions that pass the criteria test are then analyzed to identify similar components so that the total number of component type making up the configuration may be determined. The sub-graph isomorphism algorithm also identifies component type that can be prefabricated with alternative mould as an input for Optimizer unit. The sub-graph isomorphism is a computational task in which two graphs G_1 and $G_2 \subseteq \{G_i\}$ are compared to determine whether G_1 contains a sub-graph that is isomorphic to G_2 which would mean that component represented by G_2 can be produced using the same mould represented by G_1. Pairwise comparison of sub-graphs in a feasible configuration ensures that the minimum mould type can be determined for the feasible configuration S_i.

An extensive review of the graph matching algorithms has recently been made (Bunke, 1999; Cordella et al., 2004; Messmer, 1995; Ullmann, 1976). A well-known procedure for exact graph matching, reported by Ullman (1976), utilized the backtracking technique with forward checking. Ullman's algorithm suffers from the combinatory explosion problem. Messmer (1995) proposed a new approach to solve the graph matching problem for a graph database. It involves decomposing the graph into sub-graphs. A sub-graph which appears multiple times will be compared only once to the input graph (Bunke, 1999; Messmer, 1995). Cordella et al (2004) presented a graph matching algorithm, whose computational complexity is reduced using a set of feasibility rules during the matching process. The algorithm is tailored for dealing with large graphs without making particular assumptions on the nature of the graphs to be matched and can be used for both isomorphism and graph/sub-graph isomorphism. The number of building elements in a medium-size building is more than 100 so that the algorithm developed by Cordella et al (2004) has been incorporated in the proposed framework for sub-graph isomorphism analysis to reduce computational time and memory.

The principal mechanism for implementing the short-term memory function is a tabu list, or a collection of such lists, which record the configuration string of solutions (or moves) to prevent moves that lead to solutions that share subgraph in common with solutions recently rejected (i.e., generating such tabu moves).The short-term memory function, established in a tabu list, is implemented as an array *Tabu_Time(e)* recording the most recent iteration at which the infeasible component (sub-graph H_i) was created. To prevent moving back to previously investigated solutions, a tabu time *t* is defined as the time that must elapse before an edge is permitted to be moved (connected or disconnected) again, measured in number of iterations. If *Current_Time* denotes the current iteration, then a sub-graph H_i is tabu if *TabuTime(e) > Current_Time - t.* The choice of *t* is critical to the performance of our implementation of the TS algorithm. For configuration generation, $3\sqrt{n}$ (n represents number of building elements) is found empirically to be a good value for *t*. The rationale is that a tabu time larger than *n* can lead to problems in finding a non-tabu move, and the algorithm then wastes iterations without performing moves. A too short tabu time setting will lead to cycling (i.e., revisiting of previously tested problem states), and may prevent the algorithm from progressing into new (unvisited) problem states.

The described tabu search incorporates no long-term memory function and thus no long-term diversification strategy. The “short-term diversification” created by strategic oscillation was anticipated in part to compensate for this. This strategic oscillation is implemented using an infeasibility counter for orienting moves to pass feasibility borders to explore infeasible solutions (GALLEGO et al., 2011). An infeasibility counter has been incorporated to effectively traverse the nodes that may be related to the selected move. The idea behind using an infeasibility counter is to traverse an infeasible problem space in consecutive iterations, in order to make the tabu list rich so as to reduce iterations and computational time in future moves. The MaxInfCounter is set as a tradeoff between evaluating all possible connections of an individual element (6 x no. of elements) and one connection for every element in a building component (1 x no. of elements).

The components of feasible configuration may not be easy to produce and install. As can be seen in Figure 3.7, a library of preferred shapes can be used in comparator function to reduce domain of feasible configurations to a set of preferred components. The preferred components are stored in a database with the same graph format. The aforementioned algorithm is utilized to compare components of each feasible configuration against preferred components.

Finally, the comparator function sends the following results to the optimizer function for each feasible configuration: number of components type (depicted as j in optimization model), mould type that is required to precast all components (i), demand for each component in a construction cycle ($demand_j$), and ability of moulds to produce different components (ma_{ij}).

3.2.4. Optimization Models

The framework so far generates a set of feasible configuration of precast components. The next step is to determine the configuration that has the minimum cost in terms of mould fabrication, transportation and lifting. As described in section 3.1, the framework employs two evaluation criteria for evaluating the generated configurations using two optimization models namely production resource and transportation otimization. Instead, a simplified form of these models is presented in this chapter in order to show the performance and functionality of the proposed CoCoPS framework. The optimizer module presented in this chapter comprises a mathematical model to determine the number of moulds of each type that is required for each feasible configuration. Further to this, the model empirically determines the transportation and lifting cost. The optimization is at a high level planning where the number of moulds for each type is determined based on broad parameters and cycle time. The mathematical model for the optimization is built as follows (with notations depicted in Table 3.1):

Objective function: $Minimize\ z = MC + TC + LC$ (3.4)

where:

$$MC = \sum_{i=1}^{nm} Mould_i \times mfc_i + \sum_{i=1}^{nm} \left(\sum_{j=1}^{nc} \frac{comp_{j,i}}{ma_{j,i}} \right) \times \frac{NoC \times CCT}{MoL} \times mfc_i \quad (3.5)$$

$$TC = \sum_{i=1}^{nm} \sum_{j=1}^{nc} comp_{j,i} \times tc_j \quad (3.6)$$

$$LC = \sum_{i=1}^{nm} \sum_{j=1}^{nc} comp_{j,i} \times lc_j \quad (3.7)$$

Subject to following constraints:

$$\sum_{i=1}^{nm} comp_{j,i} \geq demand_j \quad \text{for } \forall j \quad (3.8)$$

$$\sum_{j=1}^{nc} \frac{comp_{j,i}}{ma_{j,i}} \leq Mould_i \times CCT \quad \text{for} \forall i \text{ and } ma_{i,j} \neq 0 \quad (3.9)$$

Transportation and lifting "tc_j" and "lc_j" are defined as:

$$tc_j = f(complexity, dimension, weight, stacking) \quad (3.10)$$

$$lc_j = f(weight, lifting-points, in-situ\ joints) \quad (3.11)$$

The objective function includes three constituents of project costs, namely: Mould Cost (*MC*), Transportation Cost (*TC*) and Lifting Cost (*LC*). Equations 3.5-3.7 represent the computation for each cost, respectively. Transportation cost of components depends on complexity of components, dimension, and weight to be stacked on truck as in Equation 3.10. Lifting cost of components also relies on weight, number lifting points and in-situ joints as in Equation 3.11. The "tc_j" and "lc_j" are calculated based on local regulations and aforementioned parameters (OSHS, 2002; Toffolon, 1984). With shape and semantic information of each component from GDM these parameters are determined empirically using the local database for prefabricated projects.

Table 3.1: Parameter and variable symbols

Parameter Symbols			
Symbol	Explanation	Symbol	Explanation
I	Index of mould type	j	Index of component type
Nm	Number of considered moulds	nc	Number of considered components
$demand_j$	Required components of type j for each cycle	$MA_{i,j}$	Ability of mould I to produce component type j
mfc_i	Mould fabrication cost of mould type i	tc_j	Transportation cost of component type j
lc_j	Lifting cost of component type j	NoC	Number of cycles
CCT	construction cycle time	MoL	Mould operational life

Variable Symbols			
MC	Total mould cost	TC	Total transportation cost
LC	Total lifting cost	$Mould_i$	Number of adopted mould of type i
$comp_{j,i}$	Component type j produced by mould type i		

Equation 3.5 yields the Mould Cost (MC) where "nm" and "nc" are number of mould types and number of component types, respectively. Different component types can be produced by different moulds represented by a matrix, $MA_{j,i}$, which is obtained from the isomorphism analysis. This concept is termed as "Component Grouping" and it will be further discussed in Chapter 5. "NoC" represents the Number of Cycles in the whole project. For example in a 10-storey building, NoC is 10 if each storey constitutes a cycle. The first constraint in Equation 3.8 depicts that the demand for components should be met. The second constraint in Equation 3.9 implies that each mould can only be used for casting once each day.

The proposed model is a Mix-IP problem. This model is solved using the CPLEX solver implemented on General Algebraic Modeling System (GAMS).

3.3. Computer Implementation

In order to demonstrate the concept of the framework presented in this study, a prototype system has been developed using C#.net platform. The preprocessor module in Figure 3.1 comprises two classes developed using the IfcSvr Active Toolbox, which is an ActiveX component that provides an interface to access IFC data model. "GetInfo" class extracts the building elements properties, geometry and topological information from IFC file to build the internal data structure.

Establishing the GDM and its related computational tasks such as configuration generation and sub-graph detection was implemented in “GraphGenerator” class. The tabu search algorithm and comparison of configurations against production and constructability criteria were undertaken by “GetFeasible” class. Finding similar components for the sake of mould optimization was implemented by the “Isomorphism” class. This class also can be used for checking components of each configuration against preferred shape components available in the library. VFLib Data-Link Library (DLL) developed by Cordella (2004) was modified to implement sub-graph isomorphism for graphs carrying attributes on nodes and edges.

The optimization in the Optimizer determines the mould usage, transportation and lifting cost that are required to minimize the total cost for each feasible configuration obtained from the Constructability Analyzer module.

3.4. Illustrative Case

The main objective of this illustrative case is to investigate the performance of using different prefabrication configurations under several demand, constraints and cost characteristics.

Although the method proposed in this study may be used for various types of buildings such as factories, infrastructures and complex buildings, for illustrative purposes, in this chapter, a residential complex comprising 20 blocks of 10 storey buildings is considered as the base scenario. Each building includes 4 spans of 5m length in each direction modelled using ArchiCAD. In this study, only columns and beams are considered among all structural and non-structural element. However, the method can be generalized to be applied for all building elements.

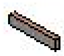
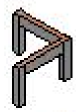
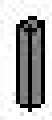
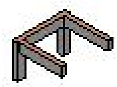
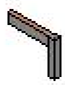
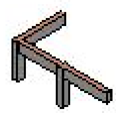
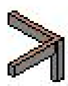
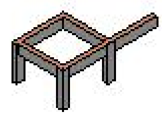
As described in the framework overview, the proposed system analyses the project in three stages. In the first stage, relevant data was retrieved from IFC. Since prefabrication of more than two storeys is infeasible (because of size constraint in transportation) the prefabrication of components that are bigger than two storeys are infeasible and thus a construction cycle is defined as prefabrication of required components for two storeys. Based on this construction cycle, 80 beams and 50 columns and their specifications including name, type, dimension, material and coordinates were extracted and stored in the database. Other required information such as area, volume, weight and length were derived. In the second stage, the topological relationships of components were extracted to make the GDM (a node-edge graph) comprising 130 vertices and 348 edges. Subsequently, all possible prefabrication configurations (S_i) were generated and compared against production and constructability criteria to obtain the feasible configurations. Local transportation constraints require that the maximum length and width of the components carried by trucks are limited to 12 m and 5 m, respectively. The maximum lifting size depends on equipment capacity. In the present study, a common lifting capacity of 20 tons is assumed. With these constraints, altogether 147 configurations met the requirements.

Finally in the third stage, the proposed mathematical model was used to optimize the mould usage, transportation and lifting cost to minimize the total cost for each feasible configuration. The cost parameters (*mfc*, *tc*, and *lc*) are derived from a cost reference table which is described as follows.

3.4.1. Cost Analysis Procedure

The three elements of cost considered are mould (fabrication and usage), transportation and lifting cost. Mould cost can range from hundreds of dollars to thousands of dollars per mould depending on mould size, complexity and materials used. Transportation and lifting cost also depend on shape, size, weight and complexity of the components as more complicated and larger components may increase the difficulties in stacking on trucks and lifting, respectively. Odd shapes are presented here to investigate the sensitivity of the model to complexity of shapes. More control on shapes is considered for real case study in Chapter 7. Moreover, a library of preferred components is introduced considering designer, manufacturer and transporter preferences. Table 3.2 summarizes the mould fabrication cost of various feasible components. These costs were derived from the precast configurations adopted in actual projects for public housing development by precast company. Prices were then adjusted for different length, number of connections and complexities for the present illustrative example. The transportation and lifting cost vary directly with the number of components, their size and shape. The costs were empirically derived from the actual precast projects database of Singapore's Housing Development Board (HDB) in Singapore. Though the costs in Table 3.2 have been adjusted to account for complexity of the precast components they are only indicative for the purpose of illustrating the process of obtaining the optimal configurations.

Table 3.2: Mould fabrication cost of precast components

Shape	Type Name	Mould Fabrication Cost (*MC*) ($)	Shape	Type Name	Mould Fabrication Cost (*MC*) ($)
	A	6,000		E	31,500
	B	5,700		F	33,000
	C	13000		G	35,000
	D	20,000		H	52,000

As can be seen in Table 3.2, multi elemental components for mould fabrication cost are more expensive in comparison with single element. But when a mould's operational life, number of construction cycles and construction cycle time (*CCT*) are considered, the lowest total cost of prefabrication comprising production, transportation and lifting may no longer be the conventional single element configuration.

To evaluate the sensitivity of the cost model to mould life, number of cycles and also construction cycle time, 27 scenarios were considered. Number of cycles (*NoC*) of 50 (10 block of 10-storey), 100 (20 block of 10-storey) and 200 (40 block of 10-storey) were considered. Mould operational life (*MoL*) depends on the material used in mould fabrication and complexity of the components. In this case study, *MoL* was varied for the typical range from 100 to 500 times per mould. An average construction cycle time (*CCT*), varying between 5 to 10 days, was also considered.

3.4.2. Result and Discussion

The optimal solution for the base scenario of 20 blocks of 10-storey (*NoC*=100, *MoL*=200, *CCT*=5) is achieved with total cost of $633,000 comprising mould cost (*MC*) of $516,000; transportation cost (*TC*) of $65,000; and lifting cost (*LC*) of $52,000. The optimal configuration is number 67 (out of 147 feasible configurations) comprising 10 Type A (single beam), 10 Type C (double-components with one beam connected to one column) and 20 Type G (three beams connected to two columns) as depicted in Table 3.2. Total number of components in each cycle is reduced from 130 individual elements to 40 components. For the production stage, 8 moulds (4 of Type G, 2 of Type A and 2 of Type C) are required to produce the necessary components for a construction cycle of 5 days. The production schedule over the 5 days is given in Figure 3.8a showing the component type using module type for each day in the cycle. Total cost is reduced by 3.8% in comparison to the conventional elemental approach.

For *CCT*=6 days, the optimal configuration is number 33 comprising 24 components of Type F, 8 components of Type A and 2 components of Type B as shown in Table 3.2 at a total cost of $585,000. In this configuration, the total number of components is 34 for each cycle. Altogether 7 moulds (4 of Type F, 1 of Type B and 1 of Type A) are required to produce the 34 components of the optimal configuration over 6 days. The production schedule is given in Figure 3.8b. The economic advantage of the optimal configuration to the conventional elemental method is 8.15%.

For *CCT*=7 days and more, the optimal configuration is number 84 comprising 2 double-component (Type C), 4 triple-component (Type D) and 14 multi-component (Type H) as depicted in Table 3.2 at a total cost of $574,000. The number of components in each cycle is reduced to 20 components. Altogether only 3 moulds (2 of Type E and 1 of Type C) are required to produce all the necessary components for cycles of 7-9 days and only 2 mould of Type E for cycle of 10 days. The production schedule over the 7-10 days is given in Figure 3.8c to 3.8f, respectively. Total cost is reduced 14.4%, 4.11%, 6.75% and 1.15% for construction cycle time of 7-10 days, respectively, in comparison to the conventional elemental approach.

Figures 3.9 and 3.10 summarize the total cost and the cost advantage of the various optimal configurations in comparison to the conventional elemental approach, respectively for the various *CCT*. The results indicate that the conventional approach (elemental prefabrication) is not an optimal solution in any of the considered scenarios. Furthermore, other scenarios were experimented with different *MoL* and *NoC*. Invariably, the optimal configurations were not changed so that the result is not sensitive to *MoL* and *NoC*.

Case #67	No. Moulds	Schedule (days)				
		1	2	3	4	5
Mould A	2	A	A	A	A	A
Mould C	2	C	C	C	C	C
Mould H	4	H	H	H	H	H

(a) CCT=5

Case #33	No. Moulds	Schedule (days)					
		1	2	3	4	5	6
Mould A	2	A	A	A	A		
Mould B	1	B	B				
Mould F	4	F	F	F	F	F	F

(b) CCT=6

Case #84	No. Moulds	Schedule (days)						
		1	2	3	4	5	6	7
Mould D	1	D	D	D	D	C	C	
Mould I	2	I	I	I	I	I	I	I

(c) CCT=7

Case #84	No. Moulds	Schedule (days)							
		1	2	3	4	5	6	7	8
Mould D	1	D	D	C	C				
Mould I	2	I	I	I	I	I	I	I	D

(d) CCT=8

Case #84	No. Moulds	Schedule (days)								
		1	2	3	4	5	6	7	8	9
Mould D	1	C	C							
Mould I	2	I	I	I	I	I	I	I	D	D

(e) CCT=9

Case #84	No. Moulds	Schedule (days)									
		1	2	3	4	5	6	7	8	9	10
Mould I	2	I	I	I	I	I	I	I	D	D	C

(f) CCT=10

Figure 3.8: Production schedule over 5-10 days cycle time

Figure 3.9 shows the variation in the total production cost for the various *CCT*. As can be seen, with an increase in the construction cycle time (5 to 10), the optimum configuration shifts from the configurations comprising the simpler components to the configurations comprising more complicated ones. For instance, for the *CCT* of 5 days, the configuration number 67 (40 components per cycle) resulted in the minimum total cost among all feasible configurations. However, when a 6 day construction cycle time was considered, configuration number 33, (34 components per cycle) is led to the minimum total cost. For the longer cycle times (7 to 10 days), the optimal configuration is number 84 with the minimum number of components (20 components per cycle). A possible explanation is that for longer construction cycle times, the moulds may be re-used to produce the simpler components in the remaining time after producing the complicated components, as depicted in the schedules of Figure 3.8 c-f. In this way, the utilization of the complex moulds is optimized to significantly reduce total cost.

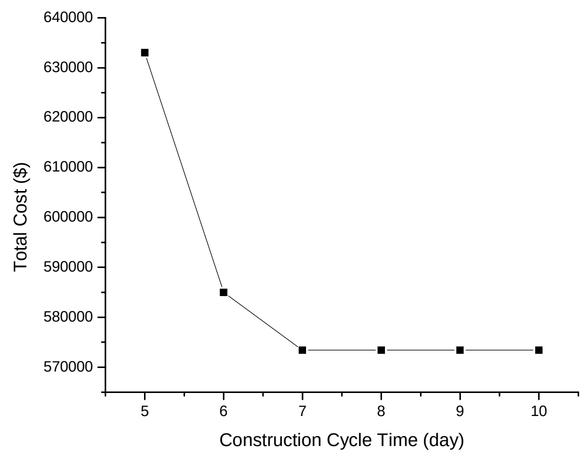


Figure 3.9: Total cost of project over construction cycle time for base scenario

Figure 3.10 shows the economic advantage of the optimal configurations in comparison with the traditional elemental prefabrication practice. As can be seen, the economic advantage of the generated optimal configurations can be as much as 14.4% depending on *CCT*. Furthermore, the results presented in Figure 3.10 reveals that the economic advantage of the generated configurations rises with an increase in construction cycle time (*CCT*) except for *CCT*= 8 and 10 days. This is because at these *CCT*s, the number of elements used in conventional configuration (80 beams and 50 columns) is divisible by the respective cycle times considered (8 and 10 days) which in turn leads to full utilization of moulds in the conventional approach. As a result, the economic advantage falls to 4.11% and 1.15% respectively.

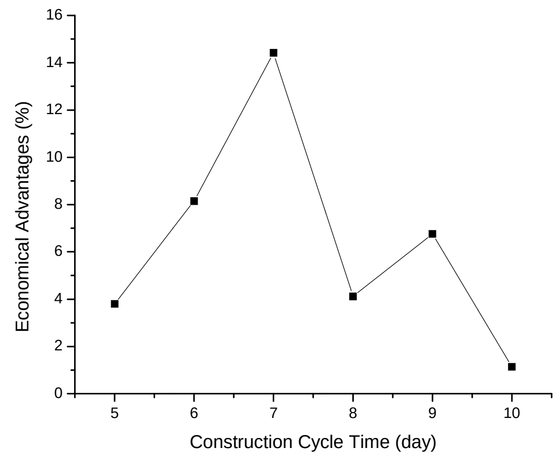


Figure 3.10: Economical advantage of optimum configuration in compare with traditional prefabrication method

3.5. Concluding Remarks

This chapter presented the concept of Component Configuration of Prefabricated Structure (CoCoPS) framework which is developed as a new approach to industrialized building system for integration between design-production coordination, standardization and prefabrication through the adoption of grouping building elements into higher level components.

Essentially, the framework is designed to extract topological relationships and geometrical properties of building elements from IFC file to be represented in a Graph Data Model (GDM) sequentially supplemented with a Geometric Network Model (GNM) from which a novel configuration string S is built. This string forms the basis by which all possible configurations can be generated using the Depth First Search (DFS) graph traversing algorithm. Feasibility is obtained by comparison of sub-graphs within the configuration string against production and constructability criteria using a modified sub-graph isomorphism algorithm. A tabu search algorithm is applied to reduce computational time in finding the set of feasible configurations. The graph isomorphism algorithm is also utilized to optimize the mould types by finding similar components in the sub-graphs within the configuration string.

The economic opportunity of the proposed framework was demonstrated by considering a residential complex of 20 blocks of 10 storey buildings. Only the prefabrication of the columns and beams were considered in the illustrative example. The results evidently showed that the proposed approach of a higher level of prefabrication may lead to reduction of total cost. Moreover, the results indicated a reduction of up to 14.4% can be achieved in comparison to the traditional elemental prefabrication. It was also seen that as the construction cycle time was increased (from 5 to 10 days), the optimum configuration shifted from those comprising the simpler components to those comprising more complicated ones. In this way, the more complex mould types can double up to produce the simpler components so that mould cost can be significantly reduced. Although complex components such as Type G and Type H (Table 3.2) may not be easily produced in a factory, preferred shape library can be a filtering method to select appropriate components based on the production and installation capabilities.

The model has taken the first step in implementing the concept of grouping building elements in prefabrication. The flexibility of the framework facilitates applying more constraints on design (for example modular coordination and vertical dimension) to obtain more practical configurations. Furthermore, although only beams and columns were considered in this chapter, the framework is able to extract require information about other building elements (e.g. slab, wall and stairs) and incorporate them into GDM and perform network analysis so that the generated configurations may result in more optimal solution for modularization. This extension is covered in Chapter 7 in detail.

It should be noted that the computation time in this model exponentially increases with number of building elements in each construction cycle. Presently, an efficient tabu search methodology of sub-graphs has been devised to avoid generating infeasible configuration strings. Further research can be planned to apply genetic algorithm techniques to further improve computational feasibility when incorporating all building elements.

CHAPTER 4: IFC-BASED GRAPH DATA MODEL (GDM) FOR TOPOLOGICAL QUERIES ON BUILDING ELEMENTS

The purpose of this chapter is to develop a new Graph Data Model (GDM) for representation and deduction of the topological relationships among 3D building elements within a building. . The GDM is utilized in CoCoPS framework to present the topological relationships among building elements. The proposed GDM is derived using weighted graph principles to simplify and abstract the topological relationships. GDM extends the existing graph-based topological models to a semantically rich model which is capable of handling various topological queries for constructability analyses. A new IFC-based algorithm for automatic deduction of relationships is proposed to form the GDM. A prototype of the proposed approach was implemented in C# platform and verified with a case study of a commercial building using two types of queries.

4.1. Introduction

Building product data model describes the physical characteristics of building elements by means of their three-dimensional (3D) geometry and topology. Geometric data represents the individual properties (e.g. location and dimension) of building elements; however, topological data denotes spatial relationships among the building elements comprising: connection, adjacency, containment, separation, and intersection. The adoption of building data models and cloud computing in construction has led to greater integration of stakeholders in the ACE/FM industry. Project participants require topological data to perform various analyses during design, production and construction processes, such as determining pedestrian navigation, indoor mapping, emergency evacuation, and clash detection. For example, architects require the adjacency and connectivity between building spaces and their boundaries to plan layout and create functional space, whereas structural engineers need information about the connections between individual structural elements to analyze structural stability. MEP engineers use the topological relationships between building elements to check compliance with building codes, for instance, sunlight analysis, heat loss, and building energy optimization. (S. Dalla Costa, 2011). Additionally, construction planners may use topological and relationship information to determine the intersection of construction zones in layout planning to detect conflicts between construction activities (Nguyen et al., 2005). These new complex tasks being incorporated into BIM applications require enhanced methods to extract and present topological relationships in 3D space, semantic information, and faster computation for performing topological queries.

A great variety of models, both 2D and 3D, deal with the topology and semantics of 3D elements within a building. However, most are developed for Geographical Information Systems (GIS) and Computer Aided Design (CAD) areas. Many researchers have extended these models in order to improve their handling of 3D geometry to perform the more complex spatial analyses used in BIM (Borrmann and Rank, 2009; Grabska et al., 2012; Paul and Borrmann, 2009; Suter and Mahdavi, 2004; Tse and Gold, 2003; Wei et al., 1998). Although these models handle most topological features, their performance is limited when running complicated queries on building elements. Most of the existing topological data models such as (Franz et al., 2005; Lamarche and Donikian, 2004; Plümer and Gröger, 1996) are geometry-driven: they handle topological relationships by geometric information and local neighborhoods. This means that the models do not explicit store the type of relationships required by a topological map used by various network-based analyses in BIM (e.g. emergency response, energy simulation or prefabrication optimization) (Rivard et al., 2000). Moreover, custom-made data tables using geometric information need to be created for every single analysis. As a result, the models are not accurate, nor efficient, nor offer reliable data storage. Inconsistent topological representation of 3D objects may arise, while slow computation time and poor data storage adversely affect the response times for real-time and cloud computing applications (e.g. evacuation routes and indoor navigation).

Further to these limitations, BIM has developed upon a semantic object-oriented approach, requiring a semantically rich representation able to handle complicated queries. A widely known standard data exchange platform for BIM exists: the Industry Foundation Classes (IFC). However, according to the literature, existing product model servers are unable to interpret the geometric and topological information that is implicitly or explicitly contained in BIM (Borrmann and Rank, 2009; Domínguez et al., 2011). Therefore the developed model for representing the topological relationships can be further extended and enriched with semantic information to be able to handle more complex topological analysis on building elements.

To bridge the limitations in the existing models, this research is inspired to exploit graph data structure to represent 3D objects and their topological relationships in 3D space and to enrich this graph with semantic information to be able to handle complicated topological queries. An elaborated Graph Data Model (GDM) is developed to map 3D objects (e.g. building elements) into a set of nodes and to convert their relationships to a set of edges using IFC as a new "topology–driven" approach. The semantic information is attached to GDM as weights of nodes and edges. The GDM is derived by using the property of graph theory and presented using graph-theoretic notations. Unlike conventional topologic models, the GDM does not start with geometrical information; it uses, instead, topological information to denote the engineering objects of a building. In the first step, the GDM is able to deduce topological information of 3D objects (both building elements and spaces) by their topological indices to set up a complete expression of building elements and spaces. In the second step, the GDM considers geometric information only to check topological relationships, effectively maintaining topological consistency within building elements. Then, topological relationship, semantic (e.g. area, volume, and material) and geometrical information is attached to the edges and nodes as a "weight," respectively, resulting in a complete description of the BIM as a weighted graph, able to carry out assorted queries with more accuracy and less computational time by using advanced graph algorithms. Further to this, the IFC serves as a universal data exchange platform and makes the model stand-alone, independent from any specific software.

In the next section of this chapter, following a deeper review of 3D topological data models, the concept and building blocks of the GDM are further refined. Then a novel methodology for automatic IFC-based deduction of topological relationships among building elements is described. Thereafter deriving the GDM is illustrated using basic elements and IFC capabilities. A computer prototype is developed to demonstrate the functionality of the proposed GDM. This prototype is evaluated using a case study in the last part of this chapter.

4.2. 3D topological data models

The topological relationships used by AEC professionals comprise adjacency, separation, containment, intersection and connectivity (Nguyen et al., 2005). Traditional topological models can be distinguished among their approaches via topology, geometry and semantics. However, there are always hybrid models using different level of abstraction (Domínguez et al., 2011). Object-based 3D data models are the topological models most relevant to the context of this research. Although these models are well established and tested for GIS applications, they are generally inadequate for implementing spatial queries within a building, because they are geometric-driven and in some cases store only limited semantic information.

The foundational schema adopted in most object-based 3D models is Boundary Representation (B-rep), such as in the TEtrahedral Network (TEN) by Pilouk (1996), the Urban Data Model (UDM) by Coors (2003), the 3D Formal Data Structure (FDS) by Molennar (1990), and the Simplified Spatial Model (SSM) by Zlatanova (2000). The B-rep has a constructive multi-layer structure in which 3D objects are described in terms of volume, face, edge, and vertex. For example, a volume is composed of faces: each face is bounded by edge loops and each edge is defined by its vertices (LaCourse, 1995).

Topological representation models have been extended for BIM by (Borrmann and Rank, 2009; Choi and Lee, 2009; Chokri and Mathieu, 2009; Laat and Berlo, 2011; Paul and Borrmann, 2009). A comprehensive review of existing models and comparison between them has been studied by Domínguez et al., (2011). The models developed by (Borrmann and Rank, 2009; Lee and Kwan, 2005; van Treeck and Rank, 2007), most closely inform the context of this paper, and are discussed comprehensively in this section.

Borrmann and Rank (2009) developed a topological model for BIM that extracts relationships in 3D space using the 9-intersection model, and implements its extraction operators by means of an octree-based algorithm. Their model explicitly represents adjacency and connectivity of 3D objects and stores ID of elements as semantic data. Thus their model is not capable of running semantic-based queries. Borrmann (2009) attempted to optimize the run time of queries according to their level of refinement. Their model requires complex geometric computation to deduce topological relationships among 3D objects in micro-spatial environments through geometric properties of 3D objects.

Both (Choi and Lee, 2009; van Treeck and Rank, 2007) use a graph-theory approach to represent geometric, topologic and semantic data of building product models. van Treeck and Rank (2007) use four different graph structures to represent: 1) geometry (graph of room face), 2) topology (structural component graph), 3) relationships (relational object graph), and 4) semantic (room graph). Although their model takes advantage of graph structure and explicitly shows the connectivity and adjacency among 3D objects, it is limited to certain building elements (wall and room) described by B-rep geometric representation. Moreover, it only maps connectivity (for the purpose of air conditioning analysis). To represent other relationships they perform complicated topological operations and graph algorithms. For example, to represent a room which contains a column, their model creates a graph with ten nodes and fourteen edges. A graph-partitioning algorithm is run to obtain the containment from this graph.

Lee and Kwan (2005) have proposed a combinatorial model, but it denotes only adjacency and connectivity of spaces within a building through a dual graph data model in order to implement the specific network analysis (Dijkstar algorithm) for emergency response analysis. Although they use a graph theoretic concept for topological representation, their model is inadequate to undertake various analysis because it is based limited topological relationships (only adjacency and connectivity), and extracts only 3D spaces (regions) from the CAD model, which are based on geometric information. Moreover, their model suffers from lack of semantic information for advanced BIM-based topological analysis.

The existing models are limited in their handling of complex semantic based queries. Most store limited semantic information about rooms, openings and walls. Further to this, with the exception of connectivity and adjacency, other topological relationships among 3D objects are not directly reflected in the models. However, BIM reuqires that the type of relationships (e.g. connectivity, containment) readily available and effectively stored to enhance the performance of spatial analysis such as energy simulation, emergency response and prefabrication optimization.

The proposed Graph Data Model (GDM) employs a graph data structure in order to overcome the limitations encountered in the existing models. It improves upon their capacity for spatial relationship analysis, by explicitly representing 3D objects and their topological relationships of connectivity, containment, intersection, and separation. The topological relationships can be better captured and represented by using the graph data model for spatial relationship analysis. The proposed GDM is a topology-driven to represent topological relationship which enhances the computational efficiency. The geometric information is used as semantic information in this model for the analytical tasks if it is required. The GDM is enriched with all semantic information that is extracted from the most widely used standard (IFC) so that it can be used by any BIM-based application. Further to this, a novel IFC-based algorithm is proposed to deduce the topological relationships among 3D objects.

4.3. The elements of the proposed Graph Data Model (GDM)

As mentioned earlier in the CoCoPS framework, the GDM simplifies, abstracts, and represents 3D objects and their relations within a building in a weighted graph. Its three main elements are node, edge and Semantic Data Table (SDT). In the GDM's graph structure, nodes represent 3D objects, and edges denote topological relationships, whose type is identified through the weights assigned to each edge. This node-edge graph is a logical network data model that can perform analyses such as Depth First Search (DFS) or shortest path. The information required for these network-based analyses is attached to nodes by a data table, the SDT, which is the third element of GDM. Having a GDM enriched with semantic information enables the users to perform advanced graph algorithms (instead of dealing with 3D objects directly) for wide ranges of topological queries, so that the computational time and data storage volume is significantly reduced. This section describes three foundation building blocks for GDM using (a) Graph data structure; (b) Graph-theoretical notations; and (c) Data base management system.

4.3.1. Graph data structure

The graph data structure denotes the pairwise topological relationships between 3D objects from a certain collection. The GDM is constructed by assigning a node to each building element (3D object) and by joining a paired node with an edge if the corresponding objects share at least a vertex. Using this transformation, building elements in actual model are mapped to nodes (or vertices) in GDM, and topological relationships among building elements are mapped to edges as shown in Figure 4.1. For example in Figure 4.1 beam B_1 and Column C_1 in a building are mapped into vertices v_1 and v_2 in the graph. Since surface S_1 is shared by objects B_1 and C_2, it is mapped into edge e_1, connecting both vertices, representing the topological relation between 3D objects B_1 and C_1. The attribute of the linking edge determines the type of relation, which is connectivity in this example.

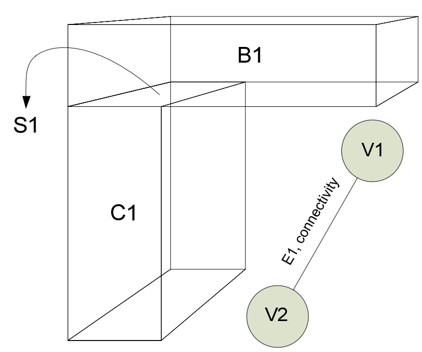


Figure 4.1: mapping 3D objects to weighted graph

The spatial relationships of the objects within an entire building are defined in a weighted graph, G. Thus, labels or weights associated with edges represent the types of topological relationships; while labels or weights assigned to vertices are the SDT which contains the properties of the 3D objects (building elements) such as geometry, size, and weight.

Figure 4.2a shows an example of a simple concrete frame comprising 8 beams, 4 columns connecting both top and bottom beam frames and one door within a wall. This concrete frame is translated to a graph using 14 nodes and 29 edges (Figure 4.2b). The type of relationship is given by the notation at edges and the properties of building elements are given by the SDT such as the attribute shown for node B1 (Figure 4.2b).

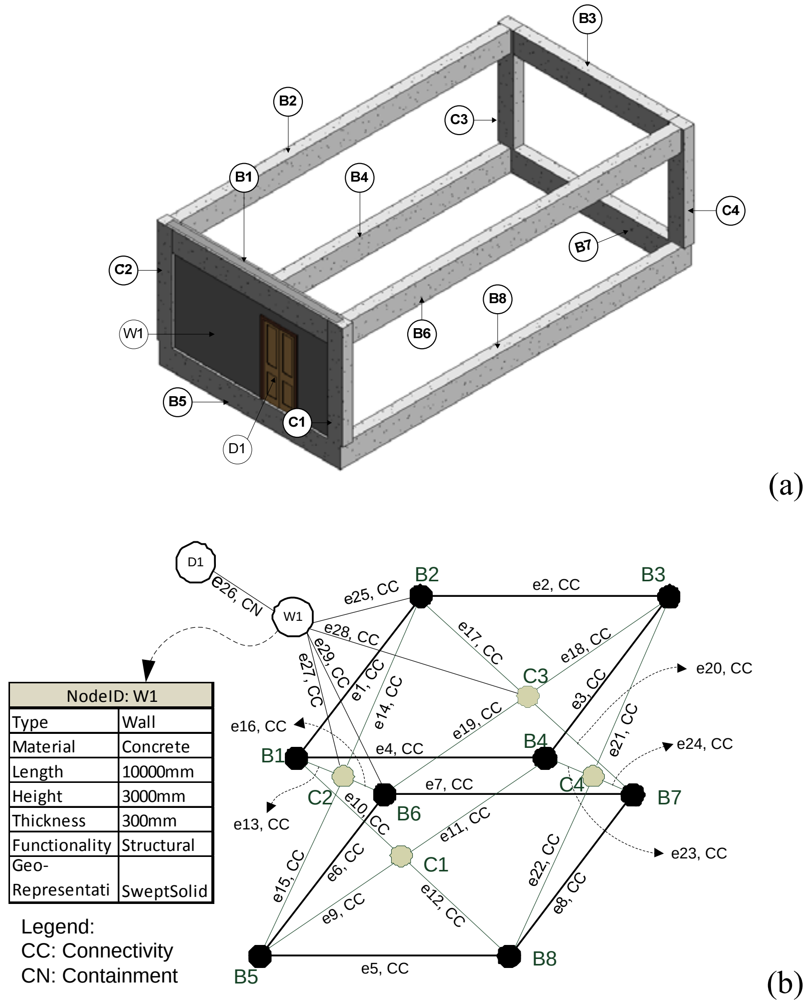


Figure 4.2: a) Sample concrete frame b) graph representation of sample frame

Using the graph data structure for representation preserves the physical and topological characteristics of building elements. Further to this, mapping the building elements and their topological relationships to a set of nodes and edges effectively maintains the topological consistency. As a result, having a graph data model enables the users to perform advanced graph algorithms (instead of dealing with 3D objects directly) for wide ranges of topological queries so that the computational time and data storage volume will be reduced significantly.

4.3.2. Graph-theory and adjacency matrix notations

The topological relationships depicted in graph G, can be represented using graph theory through specifications of a set of vertices, V, and a set of edges, E. Each edge is defined with a pair of vertices (*v, w*), where $v, w \in V$. For computational analysis and storage, the graphs are stored as a matrix, often referred to as an Adjacency Matrix (Adj).

With N building elements, the adjacency matrix is an NxN matrix. The value in Adj [*i, j*] denotes whether element *j* is an immediate successor of element *i* or not. Moreover, the value of elements of adjacency matrix comes from weights of edges in G; Adj [*i, j*] can be w_{ij} or zero as follows:

$$Adj[i,j] = \begin{cases} 0 \ \forall i = j \text{ or is not directly connected to } j \\ w_{ij} \ \forall \ i \text{ is directly connected to } j \end{cases}$$

where w_{ij} denotes the type of relationship between elements *i* and *j*.

4.3.3. Database management system

The derived model so far is a logical model or a pure graph representing relationships among 3D objects. In order to perform network-based analysis with GDM such as Depth First Search (DFS), graph isomorphism or finding shortest path additional information about the building elements may be required. To achieve this, a geometric network model together with additional information (e.g. type, material, and functionality) of 3D objects, are added to the logical network model. This additional information is carried as Semantic Data Table (SDT) in the proposed GDM. Type and detail of the information in SDT vary according to the application expected from GDM. For example for emergency response analysis, designer may just need to have room size and distance while for prefabrication, material and functionality (e.g. structural and non-structural) are also important.

To this end, relational database is employed to store the additional information and to support the graph structure by providing geometric and semantic information.

4.4. Extraction of topological and geometric data

As mentioned earlier in this chapter, the proposed GDM covers both building objects and spaces (or zones) within a building, termed as building elements. The building objects are represented as "solids" in CAD environments usually using B-rep. The information required for spatial relationship analysis are derived from the information in B-rep organized in the form of vertex, edge, face, cell, and loop. Other representation methods, such as SweptSolid, need to be transformed to B-rep in order to be used in the proposed deduction algorithm.

Figure 4.3 shows the B-Rep information structure for each primitive/feature associated with a single object associated with each primitive/features (i.e. face, edge, and vertex). The B-rep structure does not specify shared features between objects (e.g. two connected walls sharing a face). Therefore, to construct topological relationship between two objects for the GDM, the shared features must be identified. This requires, first obtaining the features comprising the boundary and interior of each object. A comparison of these features between two objects will reveal the shared features from which the associated topological relationships can be derived. This process will be elaborated in subsequent sections.

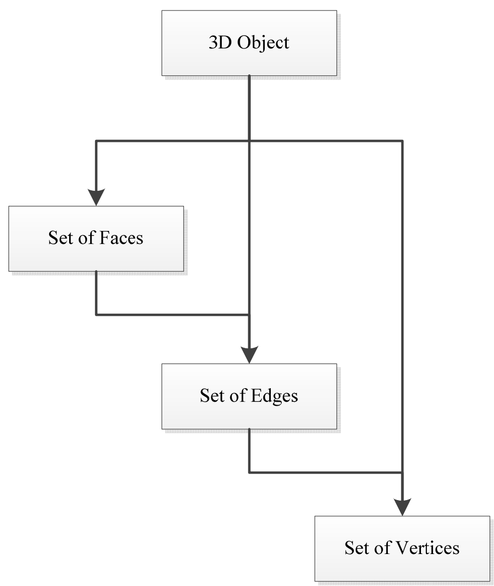


Figure 4.3: B-rep structure of a single object in association to its primitives

The emergence of new IT tools may enhance the continuous information flow which is required for data extraction. In recent years, there has been an increasing trend in the AEC in the adoption of Building Information Models (BIMs) for their processes. BIM data structure is a semantically-rich environment explicitly representing both 3D topological and geometric information, along with non-geometric properties (e.g., material properties).

The capability of extracting and exchanging data between models and applications is desirable to remedy the time consuming and error-prone process of obtaining and sharing these data. Although the proposed algorithm can be adopted for any parametric-based modeling platform, Industry Foundation Class (IFC) has been selected in this study because it is one of the most notable and widely accepted product data model in building industry. It allows efficient sharing and exchange of project information. With the IFC, it is now possible to extract topological, geometric and semantic information of building elements to complete the proposed GDM.

4.5. Topological/Geometric representation of building elements in IFC

IFC is utilized to improve the implementation of proposed topology-driven approach in deduction of topological relationships by providing the required topological features as well as predefined topological relationships in its hierarchical structure. This section presents capabilities of IFC for geometric/topological representation of 3D objects in the micro-spatial environment.

4.5.1. Representation

Basically each object that inherits from the IfcProduct class in IFC such as beam, column, wall and even a space has a geometric/topologic representation. The definitions of geometric/topologic representations in the IFC Releases 2.x are quite similar to STEP definition provided by ISO 10303- 42 (1994). Any object in IFC inheriting from

IfcProduct, has two attributes “Representation” and “ObjectPlacement”, which are the main attributes describing geometry/topology. Representation attribute defines the topological/geometry representation of an object through the properties of “IfcProductDefinitionShape”; while ObjectPlacement indicates the location of an object.

In the IFC data model, there are three main topological/geometric representations for solid modeling, namely: (1) “SweptSolid a standard geometric representation in which a profile or area is extruded along or revolved around an axis. This study is based on extrusion along an axis and it does not cover curves or circles in sweeping profile (2) Advanced geometric representation using either the CSG or the SweptSolid with enhanced profile types. The CSG representation provides a geometric representation based on the CSG model (Liebich 2004). A solid represented by CSG model is defined as a collection of primitive solids that are combined using certain operations (3) B-Rep topological model for complex geometry (Liebich 2004).

The B-rep representation is defined by IfcFacetedBrep in IFC to provide a geometric representation through a set of faces, which in turn are defined by face bounds expressed in poly loops. The lower primitives are edges and vertices. Vertices are attached to geometric data (IfcCartesianPoint) which is used for SDT. When a 3D object is defined by IfcFacetedBrep, all faces are planar and all edges are straight lines. Topological primitives are defined in IfcFacetedBrep as indicated in Figure 4.4. IFC has another entity which is used to represent 3D objects with voids. This entity is called IfcFacetedBrepWithVoids. Using these two entities together with their hierarchical structures, binary topological relationships among 3D building elements can be deducted.

In order to access the same topological features, other representations in IFC are converted to B-rep.

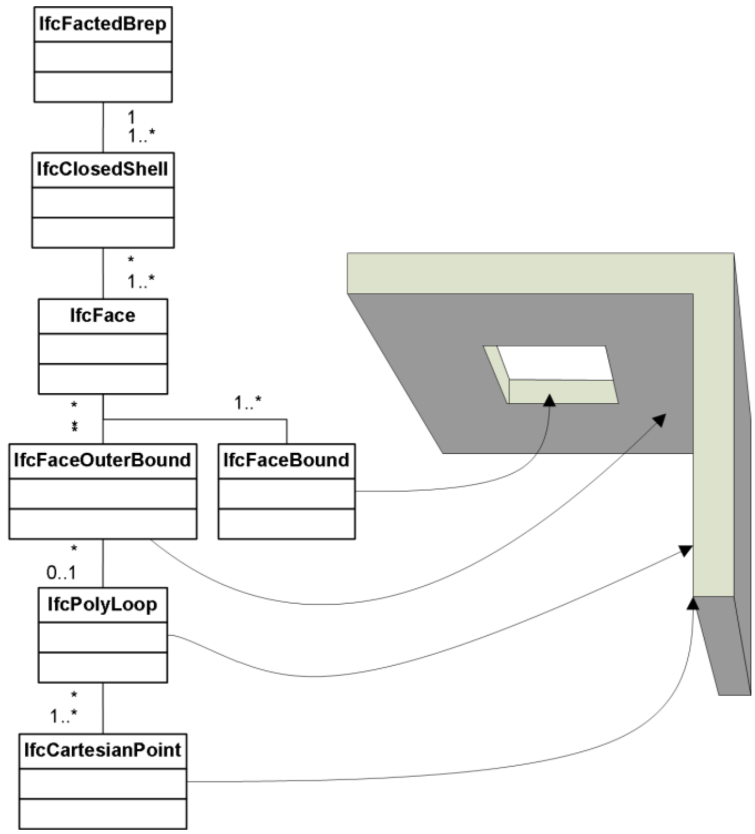


Figure 4.4: Definition of topological primitives in IFC using IfcFacetedBrep adopted from Liebich (2004)

The "SweptSolid" representation defines an object by a defined area (*IfcArbitraryProfileDef*) in a plane and an extrusion direction and length, as illustrated in Figure 4.5. It is geometry based and cannot be used directly in the proposed deduction method. An algorithm is proposed to convert "SweptSolid" representation to B-rep in order to obtain the required topological primitives through the following three steps:

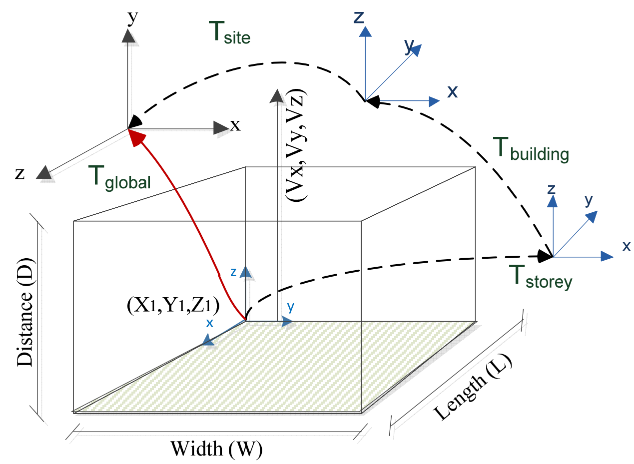


Figure 4.5: The definition of 3D geometry of a wall using a Swept Solid representation

1) Extract geometrical data of object:

Geometry of the base profile, the distance and the direction vector of extrusion together with the definition of local coordinate system are necessary information extracted from IFC file for each 3D object. For example in Figure 4.5, the extracted data include: the placement coordinate of the origin (X_1, Y_1, Z_1) of the shaded area in the local coordinate system, the length (L) and width (W) of the rectangle, direction vector of extrusion (Vx,Vy,Vz), the extrusion distance (D), and the definition of the local coordinate system of the given cube with respect to the global coordinate system.

2) Calculate coordinates in relative local system and transform to the absolute global system:

If the coordinates of vertices on the base profile is given as (x, y, z), the coordinates of the vertices on the opposite face (x', y', z') can be determined as:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + D. \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{4.1}$$

in which D denotes the extrusion distance along the vector of extrusion (V_x, V_y, V_z).

Most of the coordinates in IFC data model are presented in the relative local system. These local coordinates have to be transformed to the global coordinate system through an affine transformation (T). Using ordinary vector algebra, the global coordinates are obtained from the local coordinates in a single step (T_{Global}). Following a hierarchy of local coordinate systems (L1, L2,…, Ln), T_{Global} may be obtained by pre multiplying all local transformations (rotations and translations) as:

$$T_{Global}=T_{L1}xT_{L2}X\ldots.T_{Ln} \tag{4.2}$$

The combined local transformation matrix, T_{Local}, comprising rotation and translation with respect to the higher coordinate system is given by.

$$T_{Local} = T_{Translation} \times T_{Rotation} \Rightarrow T_{Local} = \begin{bmatrix} x_x & y_x & z_x & 0 \\ x_y & y_y & z_y & 0 \\ x_z & y_z & z_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_x & y_x & z_x & t_x \\ x_y & y_y & z_y & t_y \\ x_z & y_z & z_z & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

Typically, a wall local coordinate is referenced to a storey and local coordinate of the storey is referenced to the building. Thus the hierarchy of local coordinate system comprises T_{Wall}, T_{Storey} and $T_{Building}$ so that the transformation of the global, T_{Global}, can be obtained using Equation 4.2 with L1=wall, L2=storey and L3=Building.

3) Generate boundary faces:

Having all vertices in the global coordinate system, the boundary faces of the object can then be constructed as follows:

- Consider A_p, the base profile of the SweptSolid model, represented by an ordered set of vertices $\{V_1, V_2, ..., V_n\}$. Each vertex, V_i, has a corresponding vertex, V'_i, in the opposite face of the base profile, given by Eq. (1). Therefore the opposite side, A'_p to the base profile, is defined by $\{V'_1, V'_2, ..., V'_n\}$

- The side areas, A_i, which connects A_p and A'_p is given by the vertex set $\{V_i, V_{i+1}, V'_{i+1}, V'_i\}$ with looping (i) from 1 through n; and the value of i set to one if i = n + 1.

Using these 3 steps, a set of faces, edges, loops, and vertices are generated which will be utilized in the proposed deduction algorithm for topological relationship. Currently in IFC, no CSG primitives are included in the IFC2x specification and the use of CSG is limited to the "Clipping" representation type. In the Clipping representation, the first operand is a solid model (given by a swept area solid), the second operand is a half space solid (a 2D surface) and the Boolean operator is always DIFFERENCE. In order to convert the CSG representation to B-Rep, the previous algorithm is applied on the solid object to obtain the B-rep features. Then the intersection surface of the solid and half-space is obtained by intersecting the half-space and all side areas (Ai). The intersection lines form the edges of intersection surface.

4.5.2. Predefined topological relationships

Besides the topological/geometry representation of 3D objects, the IFC model contains predefined relationships between objects. Two topological relationships, Connectivity and assignment are two topological relationships that have been utilized to reduce the pairwise comparison of 3D objects in order to reduce computation time. These are defined in the IfcRelation entity within IfcProduct. The following describes the sub-entities of IfcRelation that have been used to deduce of topological relationships:

IfcRelConnectsElements: provides a one-to-one connectivity relationship between physical and virtual connected elements.

IfcRelConnectsPathElements: provides a one-to-one connectivity relationship between two elements, which have path information. For example wall to wall.

IfcRelContainedInSpatialStructure: assigns an element to a certain level of the spatial structure including: site, building, storey, and space. Each element can only be assigned to one level.

IfcRelFillsElement: provides a one to one relationship between an opening element and a building element that fills (or partially fills) the opening element.

IfcRelReferencedInSpatialStructure: assigns elements in addition to those levels of the project spatial structure, in which they are referenced, but not primarily contained. For example a wall might be normally assigned to only one storey, however a foundation might be assigned to particular storey and the building.

IfcRelSpaceBoundary: Defines the relationships of physical or virtual space to its surrounding elements.

IfcRelVoidsElement: Specifies the one-to-one relationship between an element and one opening element that creates a void in the element.

4.5.3. ObjectPlacement

Another necessary attribute for representation through IFC is "ObjectPlacement". Coordinate system of objects (e.g. building elements) is defined within the *ObjectPlacement*. *ObjectPlacement* of the object can either be absolute (with respect to the global coordinate system), relative (with respect to the object placement of another object), or constrained (e.g. with respect to grid axes). The default way for object placement in IFC is to use the relative placement, given by the two entities *IfcLocalPlacement* and *IfcAxis2placement3D*. *IfcLocalPlacement* defines relationships of coordinate systems recursively from higher level to lower level, whereas *IfcAxis2placement3D* supplies the position and orientation of the corresponding coordinate systems. *IfcAxis2placement3D* at the lowest level depicts the position and orientation of the object in the local coordinate system. Thus the main step in using geometrical properties of building elements is the transformation of coordinates from local to the global as given in step 2 of the algorithm presented earlier in this section.

4.6. Automatic deduction of topological relationships

Topological relationships of building elements should be deduced and assigned to edges in GDM. In order to automatically generate the GDM, an algorithm is needed to extract such topological relationships among building elements. This study develops an

IFC-based deduction algorithm for four major types of topological information comprising: Connectivity, Containment, Separation, and Intersection.

These follow the classification by Nguyen and Oloufa (2001) except that Adjacency has been omitted. In the binary relationship classified by Nguyen et al. (2005), connectivity is defined for physical building elements whereas adjacency is defined for spaces. Since both physical building elements and artificial building elements (i.e. space and workspace) are assumed as 3D objects and termed as building elements in IFC, the Adjacency and Connectivity relationships would be identical.

In the algorithm reported by Nguyen et al. (2005) the conditions of topological relationships are checked based on the extracted 3D geometric data from the CAD model. Such an approach from geometric representation and analysis of local neighborhoods is not efficient in terms of computation and data storage. Furthermore, unlike topology-driven models, geometry-driven models need geometric modifications for further analysis (Christoph van Treeck, 2003; Lam et al., 2006).

Using geometric/topological representation in IFC data model, the proposed IFC-based algorithm enhances the deduction performance in three ways. First, IFC files may be used to directly extract the topological primitives and their geometric information instead of obtaining from lines and points in a CAD environment. Second, the SweptSolid and B-rep representations of 3D objects which are supported by IFC can be used to represent a wider range of 3D objects when compared to the conventional CAD modeling. Third, IFC comprises predefined relationships between the building objects, eliminating the need for pairwise comparison to deduce binary relationships. These entities are used to reduce computation time and complexity. Indeed using IFC as a universal standard platform facilitates data exchange between different professionals so that the algorithm is independent of software.

The following describes how the classified binary relationships are identified using IFC capabilities in B-rep framework.

Connectivity:

The algorithm determines whether two specific objects share a common topological primitive (face, edge, or vertex) and the remaining vertices are outside the given object. In other words, the algorithm identifies building elements that are next-to, above, and below a given building element.

For example in Figure 4.6, when Obj1 and Obj2 share a face, there would be a unique entity (#148 IFCFACE) in their IFC topological representation which is referred by both Obj1 and Obj2. In general, to identify connectivity, the algorithm checks whether two objects referred to have the same IfcFaceOuterBound (or IfcFaceBound, IfcFace), IfcEdge and IfcVertex classes for face, edge and node connection respectively.

The algorithm then identifies whether the adjacent object is next-to, above or below the given object. This is done by forming a vector comprising two non-shared vertices of both objects. If the vector is expanded along the +z or –z axis the adjacent object is "above" and "below" respectively; otherwise it is in type of "next-to". For example in Figure 4.6, $\vec{V} = V_2 - V_1$, expands along the +z axis which implies that Obj2 is above Obj1.

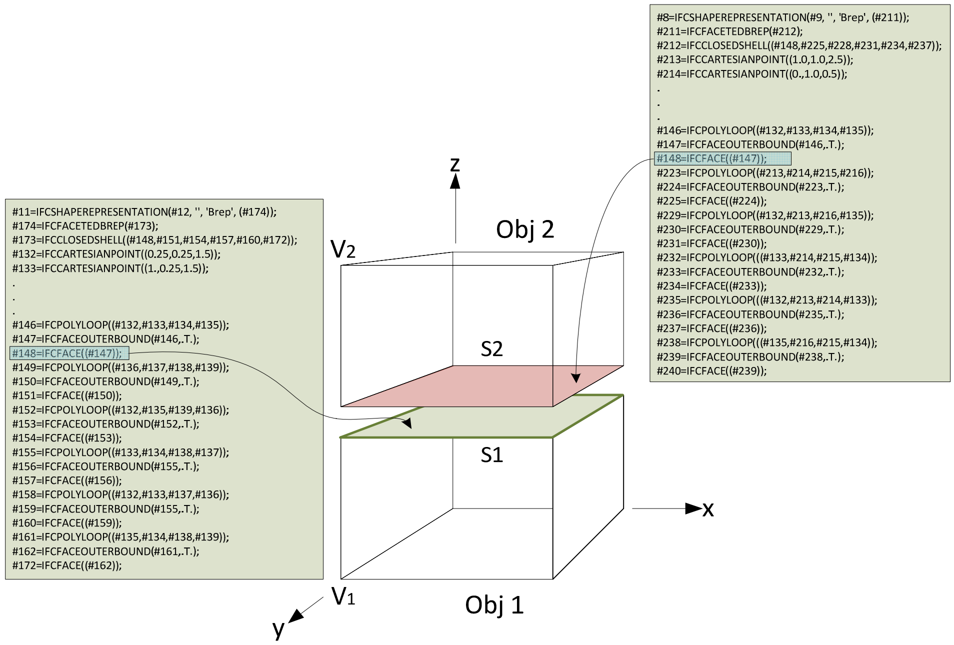


Figure 4.6: Identifying Connectivity relationship of given Obj1 and Obj2 through IFC

Containment:

There are two cases of containment of one building element by another: partially-touched and fully-contained. In containment, container object is defined by IfcFacetedBrepWithVoid entity in IFC. The IfcFacetedBrepWithVoid entity is a particular form of B-rep containing one or more voids in its interior, which are therefore represented as closed shells. The contained object is identified through this IfcClosedShell entity referred in IfcFacetedBrepWithVoid i.e. the container and the contained objects share the same IfcClosedShell. In partially-touched containment some of the vertices of the contained object are the same as container object i.e. both objects have at least one primitive in common. For example, in Figure 4.7a container Obj2 has the same IfcClosedShell (#175) as the contained object Obj1and a common edge (#251).

In the case of fully-contained, only the IfcClosedShell is common to both objects without any common primitives. For example in Figure 4.7b, boundary of Obj1 (#175, in the IFC representation) is reflected as void in the IFC representation of Obj2 (#174).

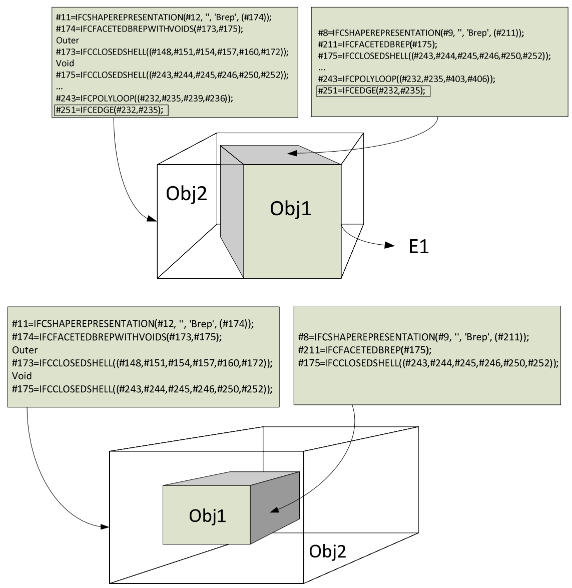


Figure 4.7: a) partially-touched containment representation in IFC, b) fully-contained topological representation in IFC

<u>Separation:</u>

When two objects are separate from each other, there is no topological primitive in common. Therefore the conditions for separation are the converse of those for containment (Figure 4.8).

To identify "Separation" relationship among building elements, the algorithm checks all the vertices of the given objects. If there is no IfcVertex or IfcCartesianPoint entity in common and both objects have the same exterior (Ellul and Haklay, 2009), then the given objects have "Separation" relationship. Figure 4.8 shows the topological representation of two separated objects (Obj1 and Obj2) in IFC. As can be seen, there is no common vertices in poly loops representing Obj1 and Obj2.

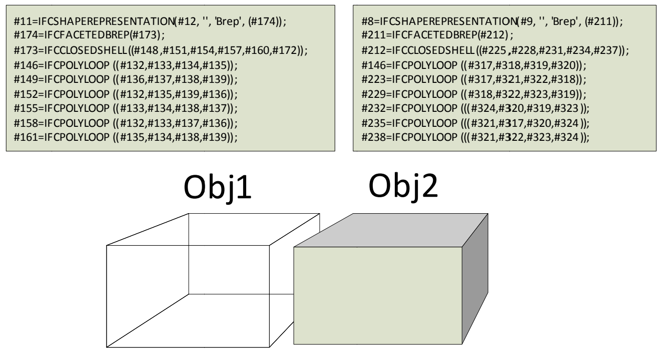


Figure 4.8: Separated objects (Obj1 and Obj2) and topological representation in IFC

Intersection:

The “Intersection” relationship identifies if two different spaces intersect one another by means of their faces or edges. Intersection has various applications in site layout planning and constructability analysis. For example, the collision of an existing building and a crane boom can be identified by means of intersection. To determine the condition of intersection between the given objects, the algorithm checks whether these objects are not contained and separate (Figure 4.9).

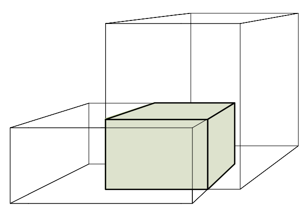


Figure 4.9: Intersection topological relationship

4.7. Deriving GDM using IFC capabilities

Figure 4.10 shows how the GDM is derived using the GDM basic elements as well as topologic/geometric representation of building elements in IFC.

The GDM is built in three stages. The first stage is to export 3D objects that are nodes in GDM from the IFC data model into a graph data structure as well as relational database. 3D objects comprise building elements (e.g. beam, column, wall or slab), spaces (e.g. room, corridor, etc.) or special zones. Information on these objects is stored in the IFC file when a CAD model is converted into an IFC file. For example IfcBeam, IfcSpace and IfcZone are entities that represent parametric data of beams, spaces and zones respectively. Required information of 3D objects that can be either identification (e.g. name, ID,...) or semantic (e.g. material, weight,…) are extracted from an IFC file so that the nodes of the GDM are obtained and stored in the nodes table in a relational database.

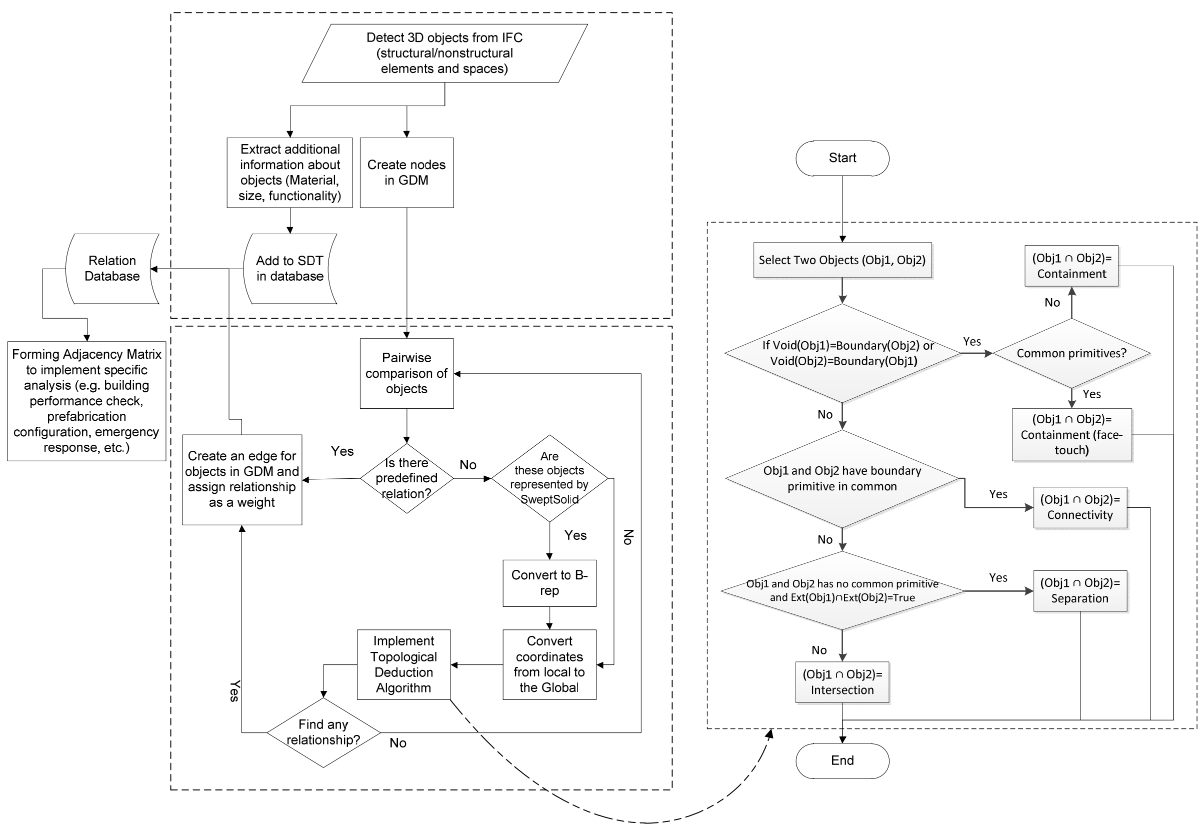


Figure 4.10: Proposed algorithm for deriving GDM

The second stage is to deduce topological relationships to be assigned as edges in GDM by pairwise comparison of objects. 3D objects may have B-rep or SweptSolid representation. B-rep primitives are utilized directly in the proposed relationship deduction algorithm. SweptSolid representation of 3D objects is converted to B-rep using the earlier three-step algorithm. Geometry data in local coordinates are transformed to the global coordinates before implementing the topological deduction algorithm. The topological deduction algorithm first checks for predefined relationships and then checks for the conditions of relationships discussed in the previous section in the following sequence: Containment, Connectivity, Separation and Intersection as presented in Figure 4.10.

The third stage forms the adjacency matrix from the relational database for network-analysis. Figure 4.11 shows the structure of the relational database comprising Nodes (representing 3D objects), Edges (representing topological relations), TopoRelType (showing relation type), SDT (semantic information), and Geometry information. Edges which correspond to the non-zero elements of adjacency matrix are obtained from the "Edges" table and the weight from the "TopoRelType" table.

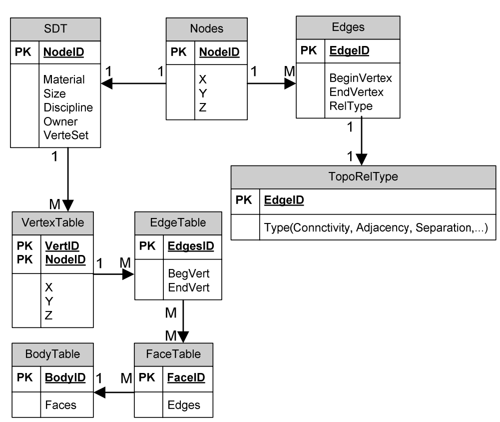


Figure 4.11: Relational database structure of proposed GDM

Suppose that the problem is to represent topological relationships among the building elements and spaces within a small building shown in Figure 4.12. The building has two rooms (SPACE 1 & 2), one corridor (SPACE 3), thirteen walls and beams, and eleven columns as labeled in Figure 4.12. Door and Windows are termed as D1, D2 and WI1 to WI3 respectively. The GDM of this building comprises 48 nodes given by V=[c1,c2, B1,B2,...]. For instance in the given building there is connectivity type relationship between “c1” and “B1” so that there is an edge, e1, in GDM with its weight assigned as “Connectivity”. The topological relationship between doors and walls for example “D1” and “W6” is “Partially-touched Containment”. There is an edge, e53, in the GDM which reflects this relationship with appropriate weight. The GDM of this building comprises 94 edges.

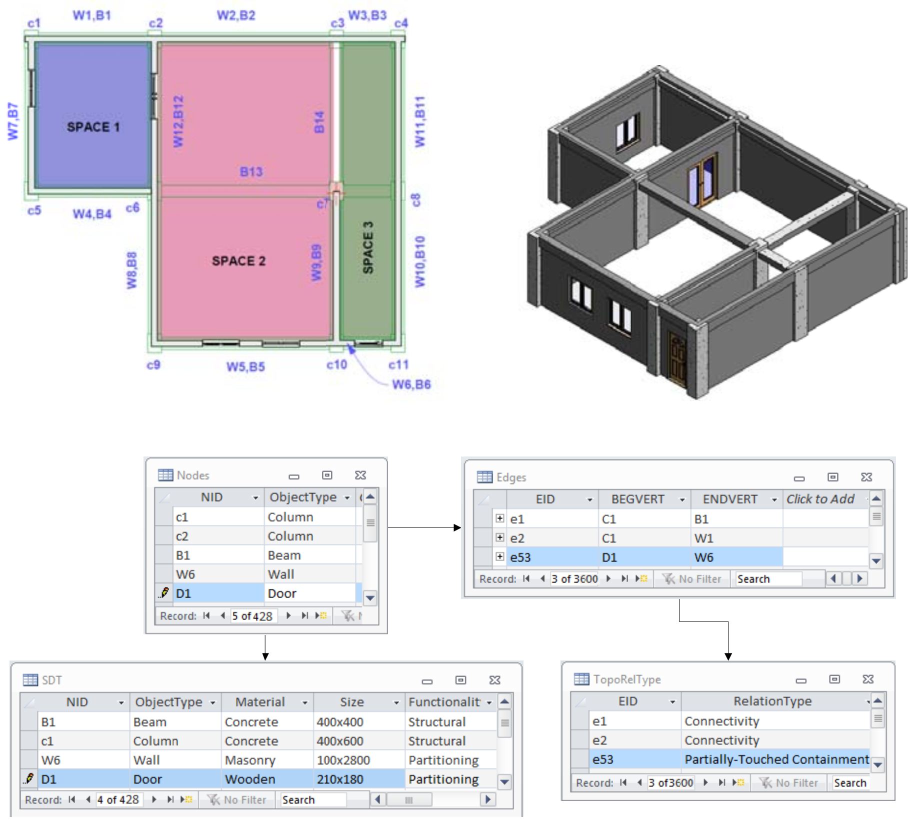


Figure 4.12: Sample 3D model, building elements and sample Relational Database structure for proposed GDM

4.8. Computer Implementation

A computer prototype system named GraphGenerator, has been implemented using C#.NET platform. The first module parses IFC to extract topological and geometric information of 3D objects. The second module checks the topological relationships among 3D objects within a building. The third module derives the GDM from the information acquired in the first two modules.

The first module comprises three classes developed using the IfcSvr Active Toolbox, which is an ActiveX component that provides an interface to access IFC data model.

“GetInfo” class in CoCoPS framework provides the building elements properties, topological identifiers and geometry information in the relational database format. This class also extracts predefined topological relationships of 3D objects (relating object, related object). “ConvertLocaltoGlobal” class converts local geometry of building elements to the global coordinate system. “SweptToBrep” class performs the 3-step algorithm described earlier in this chapterr to transform SweptSolid representation to B-rep representation. The second module includes a class which is named “CheckRel”. This class traverses all the topological identifiers of each object to find out current position of the object in relation to others, thus establishing the topological relationships. The third module employs “DeriveGDM” class with “getNodes()”, “getEdges()” and “getSDT()” functions to derive the corresponding three elements of the GDM. Finally, the output from the prototype can be either shown as text report or stored in the database for further analysis or queries.

4.9. Case Study

To examine the potential benefits of the GDM as well as the IFC-based deduction algorithm presented in this chapter, an experimental implementation was conducted. Basically the case is targeted to investigate the feasibility of using IFC platform to deduce topological relationships. It is designed to calculate the accuracy of the proposed algorithm in identification of all defined topological relationships. The accuracy check presents the relative error of the proposed model in detecting all existing relationships. Furthermore, this case study evaluates the efficiencies of spatial queries in terms of time complexity using the GDM in comparison with ISO 19107 spatial schema which is commonly used in conventional models (Seokho, 2010)

ISO19107 Spatial Schema uses topology primitives (Node, Edge, Face, and Solid) to define spatial objects. In this schema, 3D object is expressed in combination of face boundaries, edges and vertices hierarchically. Coordinates that are used to represent geometry location are defined in node primitive.

A five-storey commercial building located in Singapore was used as a study site. Figure 4.13 shows a section of one of its floors. There are over thirty 3D spaces such as “office”, “hall”, “lift box”, and “corridor” in each storey. The building elements included in the case model are 46 columns, 114 beams, 138 walls, and 21 doors. The spaces in different storeys are connected through the staircases so that there are altogether 165 3D spaces for the whole building. This model was created by Autodesk Revit Structure and exported to an IFC format.

In most of the geometric driven models, such as (Borrmann and Rank 2009), accuracy is measured based on refinement level (size of local neighborhood). The optimal level of refinement determines the accuracy of the model (Borrmann and Rank, 2009). However, the proposed IFC-based algorithm is a deterministic model: it may not identify an existing topological relationship because of its representation format in IFC. In other words, the level of refinement and runtime is not applied for an accuracy check. Since the algorithm is based on pairwise check, the runtime depends only on the number of building elements.

The GDM's accuracy is measured by relative errors for identification of following topological relationships in the case study: 1) Connectivity (Space-Space, Building element-Building element), 2) Containment (Partially-Touched, Fully-Contained), 3) Separation, and 4) Intersection.

Results indicate that the model can capture the connectivity of spaces completely (0% relative error). However, the connectivity of building elements has a 15% relative error. The relative error consists of 11% missing connections and 4% extra detected connections. The containment accuracy check explained these extra-detected connections. Missing connections reveal that the algorithm could not detect 376 connections out of 3420 existing connections. From these missing connections, 29 (<1%) belong to curve shape elements which have different representations from the ones covered. The remaining missing connections (347) are for building elements that have no face, edge, or node in common. In this case the first object (with a smaller cross-section) is connected to the middle of one face of the second object. These errors can be reduced by either including more representation methods or checking transitivity further to pairwise. For example, if A is connected to B and B is connected to C, then check the primitives of A and C for possible connections.

Accuracy is checked for both fully-contained and partially-touched. The algorithm successfully detected all fully-contained relationships (0% relative error). However, there is 5.6% relative error in partially-touched detection. A detailed inspection revealed that if the contained object has one or more face in touch with the container, the relationship is considered as connectivity.

The relative error of detecting separation among building elements is 0.08%. It shows that 376 extra separations have been detected out of 47,023. These extra separations have been detected because the algorithm could not detect the connectivity.

The results of intersection detection show a 23% relative error. The algorithm could not identify 19 intersections out of 83 existing intersections. This is because of the complicated topological representation that IFC uses due to irregular shape of 3D spaces. However, the algorithm could successfully identify the intersection of lift shaft with all three floors separately.

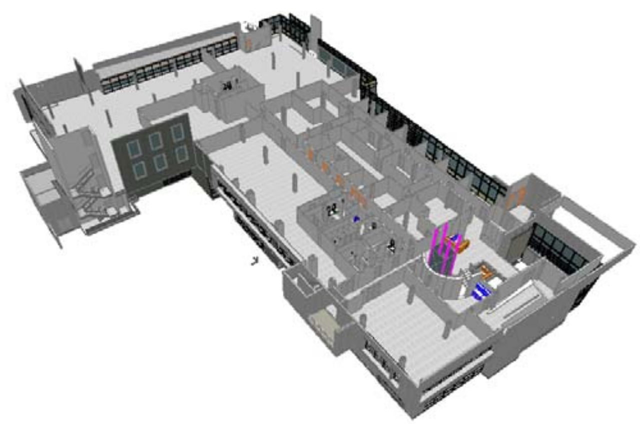


Figure 4.13: Five-storey commercial building located in Singapore selected as study site

Figure 4.14 shows the sample output from the prototype for the given case study indicating the spatial relationships between 3D objects. The deduced topological relationships can be stored in a database and retrieved for further analysis.

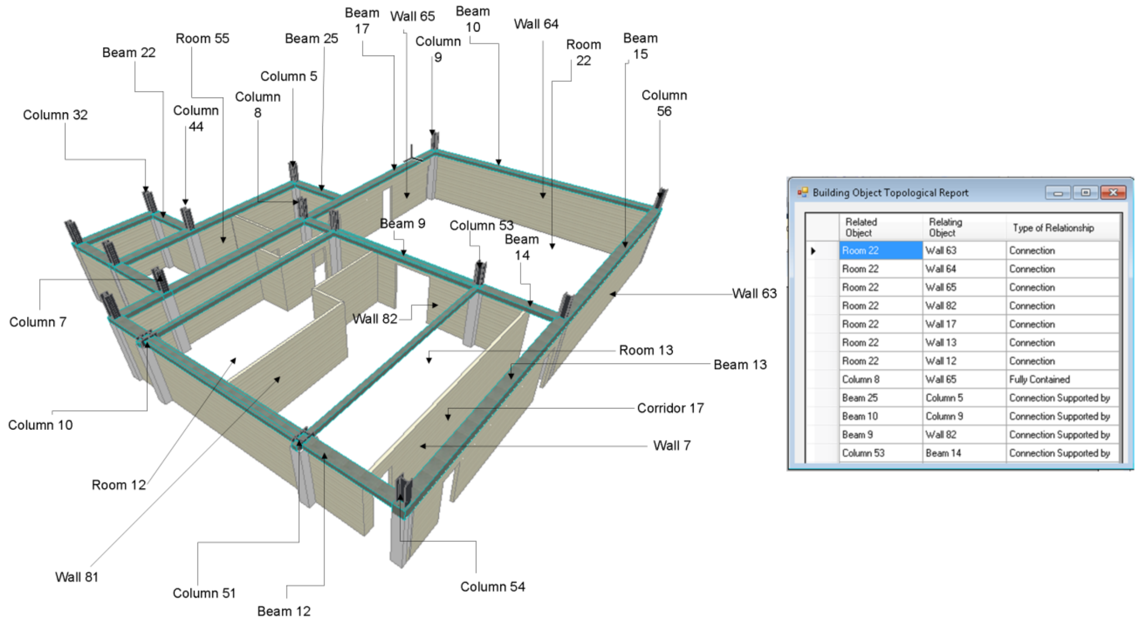


Figure 4.14: Sample output from developed prototype for given study site

In order to evaluate the time complexity of queries on both conventional ISO 19107 data model and the GDM, two queries are performed. The first is the connectivity detection to find the shortest path for emergency response. The second query is to find the similar component within a structure that can be prefabricated with the same mold. This query will require additional isomorphism analysis.

Required data based on ISO 19107 Spatial schema is stored in a Relational Database (RD). Since the RD of existing model (ISO 19107) in general is inadequate to enable the user to implement all types of queries, it should be customized for each particular purpose.

Query 1: "Connectivity" detection for emergency response

The first query is conducted to find the shortest path from the targeted room to exit for emergency response analysis. The algorithm and relational database used by Seokho (2010) have been adopted for the query. In their model, each Node, Edge, Face and 3DObject tables have 'many-to-many' relationships that should be normalized (3rd normal form) before executing queries. The query is performed in the following sequence:

1) Select a target space [S_0] and find the faces with door/s [F_1, F_2,…] composing the space [S_0].

2) Find the spaces [S_i] that shares the same faces [F_1, F_2,…].

3) If the exit area is in [S_i] then stop; otherwise

4) Find the faces that include door/s [F'_1, F'_2,…] that compose the spaces [S_i].

5) Repeat steps 2 to 4 to find the connected spaces to the exit area.

In addition, the query needs to check all the distinguished paths to find the shortest one according to the geometry information.

The process of finding emergency path on GDM is performed using a weighted shortest path algorithm as follows:

1) Select a target space represented as [N_0] and the exit node [E] in the GDM.

2) Perform shortest path algorithm (Dijkstra) to establish the cheapest exit route with the following detail:

3) Find the connected spaces through one of the following ways:

a) Predefined topological relation (the edge connects two space with the "Connectivity" weight)

b) Walls connecting two spaces with a door in common

4) Apply the cost of the shortest path algorithm (Dijkstra) which is the centre to centre distance between connected spaces obtained from SDT.

Figure 4.15 presents the processing time of the emergency response query using the two data models with the x-axis denoting the number of connected spaces in the detected shortest path. The query times between the two data models start to show a significant difference when there are more than 30 3D spaces. With 100 objects, it takes 64 seconds using conventional ISO 19107 spatial schema whereas it takes 15 seconds using GDM. The computation cost for conventional ISO 19107 approach increases exponentially to find the shortest exit route of farther spaces. On the other hand, the computation using GDM does not show any exponential increase with spaces. Thus GDM is much more efficient than the ISO 19107 Spatial Schema for connectivity queries.

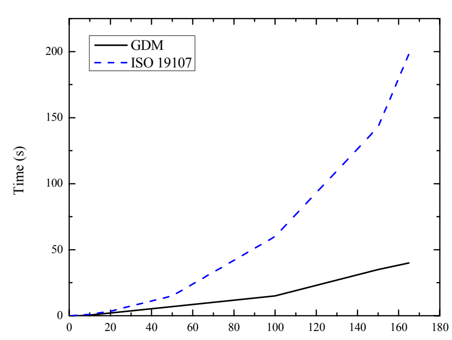


Figure 4.15: Time vs. Number of 3D spaces of emergency response query for GDM and ISO 19107 Spatial Schema

Query 2: "Prefabrication Configuration" query for mould optimization

The second query is designed to demonstrate the strength of the proposed GDM in the context of this study. Therefore, the second query is aimed to find similar components for constructability analyzer module of CoCoPS framework that was described in chapter 3. The similar components are casted with the same mould to reduce prefabrication cost and increase the constructability of precast structures. This query is performed in two parts. The first part detects connected building components. This is followed by executing weighted graph isomorphism algorithm to find similar components in the second part.

The first part of the query is implemented using both ISO 19107 and GDM; however the second part of the query can only be implemented on graph models. Figure 4.16 depicts the result of the second query. Figure 4.16a shows the time taken to detect connectivity of building elements using the two data models. The query times rarely seem to have differences between the two data models until 20 building components. When connectivity query is run for 100 or more components, the time using conventional data model takes over 4 times more than the time using GDM. Since the algorithm relies on pairwise comparison of 3D objects, detection time exponentially increases for more than 800 elements; however the detection time for the proposed GDM remains linear.

Graph isomorphism can be conducted only on graph models so that the result for the second part of the query is presented only for GDM in Figure 4.16b. Despite the complexity of the query such as graph isomorphism and the size of the problem, the computational effort remains feasible using the proposed GDM approach

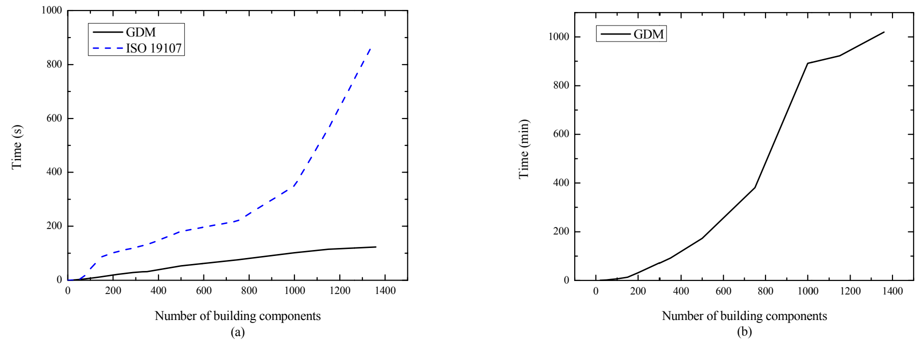


Figure 4.16: a) "Connectivity" detection time among building elements. b) Time for finding similar components within a building using the GDM

4.10. Concluding Remarks

This study develops a semantic-rich, 3D topological data model to represent the topological relationships among 3D objects in buildings. The 3D data model is derived using weighted graph elements, and formulated with graph-theory and adjacency matrix notations. This 3D topological data model is called the Graph Data Model (GDM). The GDM exploits IFC capabilities for geometric/topological representation, thus simplifying the abstraction of the topological relationships among 3D objects using the node-edge structure of the graph. The semantic information is added as weights to nodes and edges and is termed SDT. The accuracy check for the proposed model shows 15%, 5.6%, 0.08%, and 23% relative error in detection of connectivity, containment, separation, and intersection respectively. The comparison of running two sample analysis using both conventional ISO 19107 spatial schema and GDM demonstrates the capability of the proposed GDM to handle complex queries efficiently.

The elements of the proposed GDM make it an elaborated intelligent model to represent topological relationships that will be able to handle wide ranges of queries efficiently. Having a weighted graph data structure, complex topological queries can be implemented through advanced graph algorithms. Moreover SDT makes the GDM a knowledge embedded model which would be able to run rule-based queries and constraints such as weighted graph isomorphism algorithm required in query 2.

The GDM presented in this chapter contributes to the advancement of research in the area of 3D topological models for AEC applications and also overcomes several limitations of the existing models in following ways: first, it explicitly represents the elements (structural and non-structural) of buildings using a weighted graph data structure; second, since the proposed model is not limited to any specific geometric representation of 3D objects such as B-rep, SweptSolid, or CSG a wide range of 3D objects can be modeled. However, the proposed model does not cover curved-shape building elements. The model will be extended to cover circle and curved-shape elements in the future work. Third, using IFC as a data exchange platform enables the GDM to exploit the pre-defined topological relationships in IFC to significantly reduce the deduction time. Fourth, previous graph data models such as those developed by Borrmann (2009), Van Treeck (2009) and Lee and Kwan (2005), are limited to adjacency and connectivity, for the sake of specific purpose of finding shortest path among spaces in a building, whereas the GDM covers all four major spatial relationships for all AEC/FM applications. Fifth, the proposed GDM is enriched with the semantic information obtained from IFC, so that it is capable of handling complex semantic based queries for different project stakeholders. Sixth, and finally, network-based analyses can be performed to maintain computational efficiency, while avoiding storage of massive geometric data of complex 3D objects.

The GDM is employed in the constructability analyzer module of the proposed CoCoPS framework to represent the topological relationships of building elements. The Depth First Search (DFS) algorithm is implemented on GDM to find building components. Then after the advanced sub-graph algorithm is performed on the results to detect similar components. The performance of the GDM was examined for DFS and graph isomorphism queries separately over through the case studies in this chapter; however, the result for combined query is presented in the chapter 7.

CHAPTER 5: INTEGRATED PREFABRICATION CONFIGURATION AND COMPONENT GROUPING FOR OPTIMIZATION OF PRECAST PRODUCTION PLANING

This chapter provides an integrated plan to manage available resources in an appropriate way to produce prefabricated components to be able to satisfy design flexibility, production constraints and installation demands. With a higher degree of prefabrication more complex moulds are required. In relation to this, the idea of prefabrication configuration is integrated with a new idea, namely component grouping to optimize precast production resource and cost. Based on these concepts, a MIP optimization model is developed to adopt appropriate moulds and create the optimal production plan. The model is validated using two examples with different scenarios.

5.1. Introduction

The optimization on prefabrication planning has been studied since 1979 from various perspectives (Dawood, 1995; Huang et al., 2005). Several planning and scheduling models have been developed and optimized specifically for precast concrete structures using various methods such as mathematical programming (Chan and Hu, 2002a; Warszawski, 1984), simulation (Balbontin-Bravo, 1998), capacity-planning (Dawood and Neale, 1993), and process scheduling (Leu and Hwang, 2002).

Operational planning and scheduling in multi-product, multi-stage plants (such as precast factories) is inherently complex and is usually planned over a horizon of 20 over years (Warszawski and Ishai, 1982). The main considerations for planners in such plants are resources such as mould usage and frequent mould changeovers.

The resource planning problem in prefabrication has rarely been considered in previous studies. Leu and Hwang (2002) developed a resource-constrained flow-shop scheduling model for precast production which is solved by a genetic algorithm. The constraints considered in their model include labour, cranes, reinforcement cage storage, and curing capacity. Mould planning is not included as a resource or constraint in this scheduling model. However, studies and surveys on the resource and planning optimization of precast elements reveals that the main equipment in a prefabrication plants are casting moulds (Chan and Hu, 2002a; Hao, 2007; Huang et al., 2005; Zhai et al., 2008). In addition, the model developed by Leu and Hwang (2002) can only be applied to simple production situations.

In these prefabrication planning models, building elements are assumed to be produced individually. However, several types of precast elements may be produced on the same mould group with slight variations. This is called the grouping concept Huang et al. (2005).

The grouping concept becomes important when a higher degree of prefabrication is employed. In this approach, more complex moulds are required to produce prefabricated components. With the grouping concept, the complex moulds can also be utilized to produce smaller components. However, this has to be carefully strategized to ensure optimal utilization. Therefore in production planning and resource optimization, the number and the type of mould needed for production must be determined.

Mould numbers and types are limited due to their cost and the available space in production plants. Furthermore, frequent mould changeovers to produce different precast components are costly and time consuming. Therefore, a decision on mould types and numbers that also meets the demand schedule is significant to production costs.

To this end, the objective of this chapter is to develop a Mixed Integer Programming (MIP) model for the integrated problem of operational planning and mould allocation in precast production when a higher degree of prefabrication with more complex components is adopted. A new approach is presented that utilizes the idea of prefabrication configuration and component grouping for resource optimization in precast plants. The model develops a production schedule to meet the construction site demands while also considering prefab component complexity, production resources (mould planning) and factory constraints (e.g. material supply, production area). The results are validated using two case studies with different scenarios.

5.2. Prefabrication Configuration and Component Grouping Concepts

Developing the proposed optimization model requires the use of “Prefabrication Configuration”, “Component Type”, “Mould Type”, and “Component Group”. This section elaborates on these concepts and explains how they fit into the optimization model.

5.2.1. Prefabrication Configuration:

As described earlier in this study, the configuration of the elements in a building that is to be produced off-site is called the “Prefabrication Configuration”. Figure 5.1a shows the floor plan of a prefabrication project and Figure 5.1b shows its 3D view. Traditionally, building elements are produced individually in a production plant and assembled at the construction site, as shown in Figure 5.1c. However, these elements can be grouped and a new arrangement of components can be established for the prefabrication of the same plan, as shown in Figure 5.1d. This is the basis of the “higher degree of prefabrication” that was addressed earlier in this chapter. Prefabrication configuration may range from small structural components at level 2 (refer to Figure 1.1) to building systems or modules at level 4. Each prefabrication configuration requires certain moulds and resources in the production plant. The number of configurations for prefabrication increases by the power of the number of building elements. Most configurations are not feasible due to design, production, transportation or construction constraints. The set of feasible configurations are provided by Graph Generator and Constructability Analyzer in CoCoPS framework.

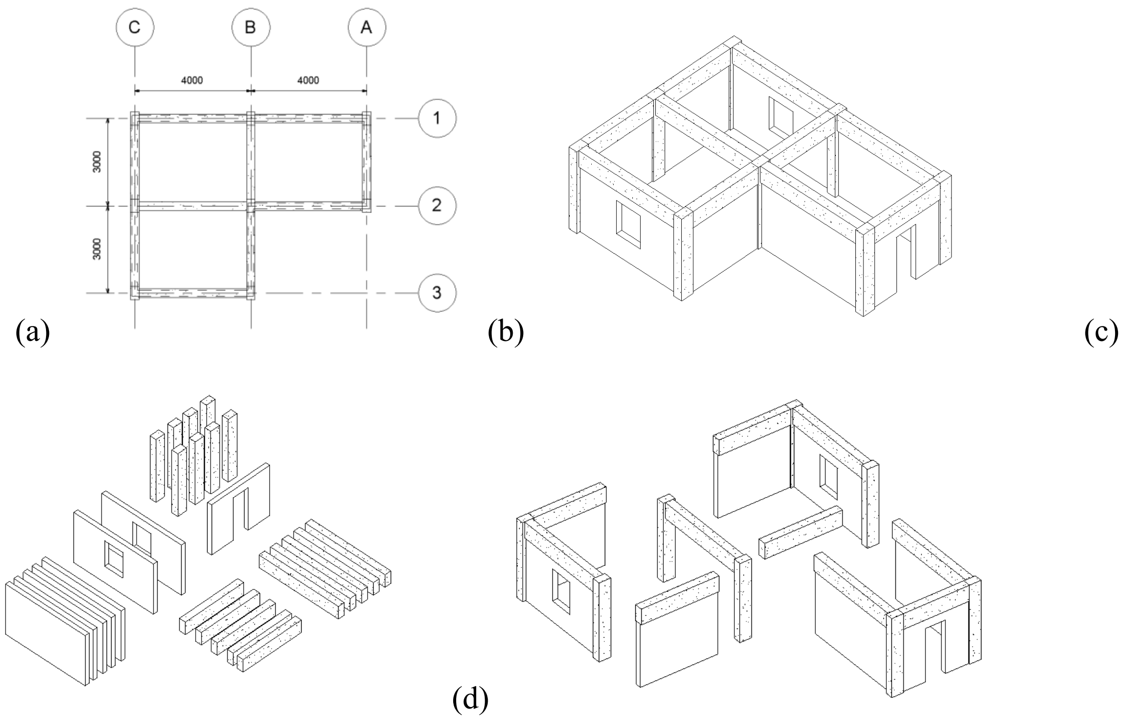
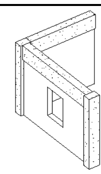
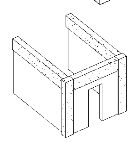
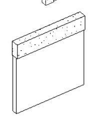
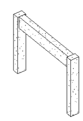
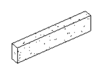


Figure 5.1: a- Sample building plan, b- Sample building 3D view, c- building elements for conventional prefabrication, d- building components for a sample configuration of building

5.2.2. Component Type

Each identical component in a prefabrication configuration is designated as a "Component Type". Components are standardized into types based on their shape, size, material, strength and functionality. For example, in the plan given in Figure 5.1, there are six types of components comprising two types of beams, three types of walls and one type of column in the traditional elemental configuration (Figure 5.1c). The other configuration shown in Figure 5.1d comprises five component types, as indicated in Table 5.1.

Table 5.1: List of component type of sample configuration in Figure 5.1

Component Type	Shape	Quantity
1		2
2		1
3		1
4		1
5		1

5.2.3. Mould Type

In most factories production work can be unified. Steel moulds with high initial costs are used to produce components smoothly and repeatedly. For each component type there is one "Mould Type". However, moulds can be used to produce a range of components with small variations in shape and size to increase standardization and mould utilization in production. For example, in the given plan (Figure 5.1), the traditional prefabrication configuration initially requires six mould types to produce all of the component types. Since there are two straight beam types that only differ in length, one mould type may be able to produce both beam types (Figure 5.2). This idea can be implemented on more complex moulds to produce a variety of smaller component types at the same time by using partitioning and separators. In this way, the utilization of moulds becomes a key issue in planning optimization. This is further elaborated in the following paragraph on component grouping. The type and the number of each mould in the production plant should be determined in a way that satisfies construction requirements, including schedule demand, with minimum costs.

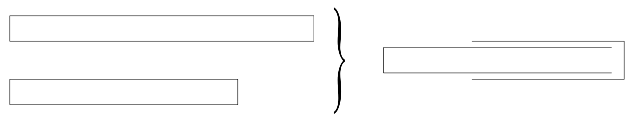


Figure 5.2: Straight beam mould can produce range of beam size

5.2.4. Component Group

When a high degree of prefabrication is employed to produce building components it may include complex and irregular-shaped component types, which in turn require complex moulds. In addition, moulds must be replaced with new ones after a certain number of castings (mould life cycle). Therefore, efforts should be made to fully utilize moulds during their life cycle. In order to increase the degree of standardization and resource utilization, complex mould types can be used to produce smaller components through partitioning and other practical techniques. To formulate this concept, the "Component Group" is defined as the set of component types that can be produced with one particular mould in a single casting cycle.

As shown in Figure 5.3, for the prefabrication configuration described in Figure 5.1d, the mould type required for component type 1 (see Table 5.1) may be used to produce either one component type 1 (denoted component group 1) or one component type 3 with one component type 4 (together denoted as component group 3) in one casting.

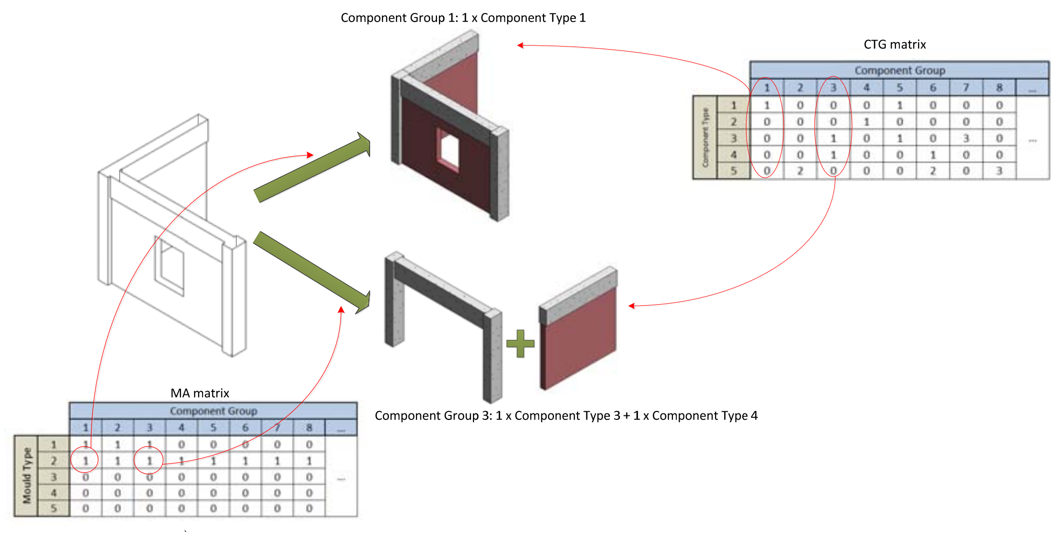


Figure 5.3: Sample component groups producing by mould type I.

As shown in Figure 5.4, the relationship between “Component Type” and “Mould Type” is a many-to-many relationship and can be normalized through the component grouping concept using two matrices.

The first matrix, *CTG*, maps the composition of component types in a component group. The CoCoPS framework calculates the *CTG* matrix for each prefabrication configuration using sub-graph isomorphism analysis. The second matrix is a binary matrix denoting the “Mould Adaptability” (*MA*). It describes the capability of each mould type to produce different component groups in which a “1” denotes that the component group may be produced by the corresponding mould type. Taking Figure 5.3 as an example, mould type 1 may be used to produce one of component groups 1, 2 or 3 in a single casting. From the component group perspective, component 1 may be produced by mould type 1 or 2 (See Table 5.1 for the component types).

Mould changeover occurs when moulds produce different component groups in two succeeding production cycles (or casting cycles). Mould changeover is time-consuming and incurs additional costs for production.

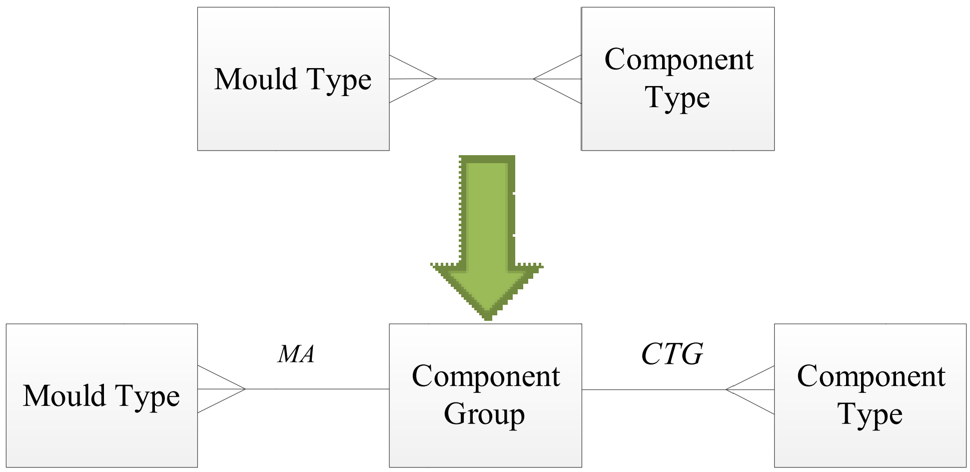


Figure 5.4: The many-to-many relationship of component type and mould type

5.3. Problem Statement

Suppose precast manufacturing facility (**F**) receives a feasible configuration of building components and must decide the optimal plan in terms of mould utilization. The configuration consists of J component types ($j=1, 2,\dots, J$). Each component type (j) may be produced using a dedicated mould type (i) so that the total number of dedicated mould types (i) is $I=J$. Producer (**F**) may be required to employ several moulds of each mould type to meet the construction demands. Thus, a new index, (l) ($l=1,2,\dots,nl$), is defined to address instances of each mould type in which nl is the maximum number of moulds for each type. The index of (h) ($h=1,2,\dots,H$) is used to indicate each component group. The total number of component groups in this configuration is (H).

Figure 5.5 presents the definitions of planning horizon, casting cycle, delivery date and construction cycle as used in this study. The production and delivery of components are planned for each construction cycle (t) ($t=1, 2, \ldots T$). A construction cycle is defined as the prefabrication of required components for a certain number of storeys with the same design (repetitive floor plan). The project plan (P) is divided into T uniform intervals of construction cycles.

The construction cycle is segmented into K workdays to accommodate the production or casting cycle (k). The production or casting cycle (k) is defined as the unit of time it takes to complete all production tasks including mould preparation, reinforcing, casting, curing, finishing and transportation to the production stockyard. The production or casting cycle is assumed as one workday.

For a given configuration, there is a demand profile with pre-specified due dates or delivery dates. The construction demand according to the installation sequence is denoted by $DD_{j,k,t}$, indicating the number of components of type (j) to be delivered on the k^{th} day of construction cycle t.

The concepts of mould type, component type and component group are included in Figure 5.5 to show their relationship to the production cycle. For example, Figure 5.5 shows four moulds comprising of one mould of type 1 and three moulds of type 2 being adopted to produce all of the required components. Thus, the index of the mould instance is l=1 for mould type 1 (i=1), while it ranges from 1 to 3 (l=1,2,3) for mould type 2 (i=2). The mould adaptability matrix ($MA_{i,h}$) in Figure 5.6 shows that mould type 2 is an l-shaped mould that can produce either l-shaped components or two straight components, which are named component groups 4 and 5 respectively. The third instance (*l*=3) of mould type 2 (*i*=1) produces component group 4 (*h*=4) on the 5th working day (*k*=5) of construction cycle 8 (*t*=8).

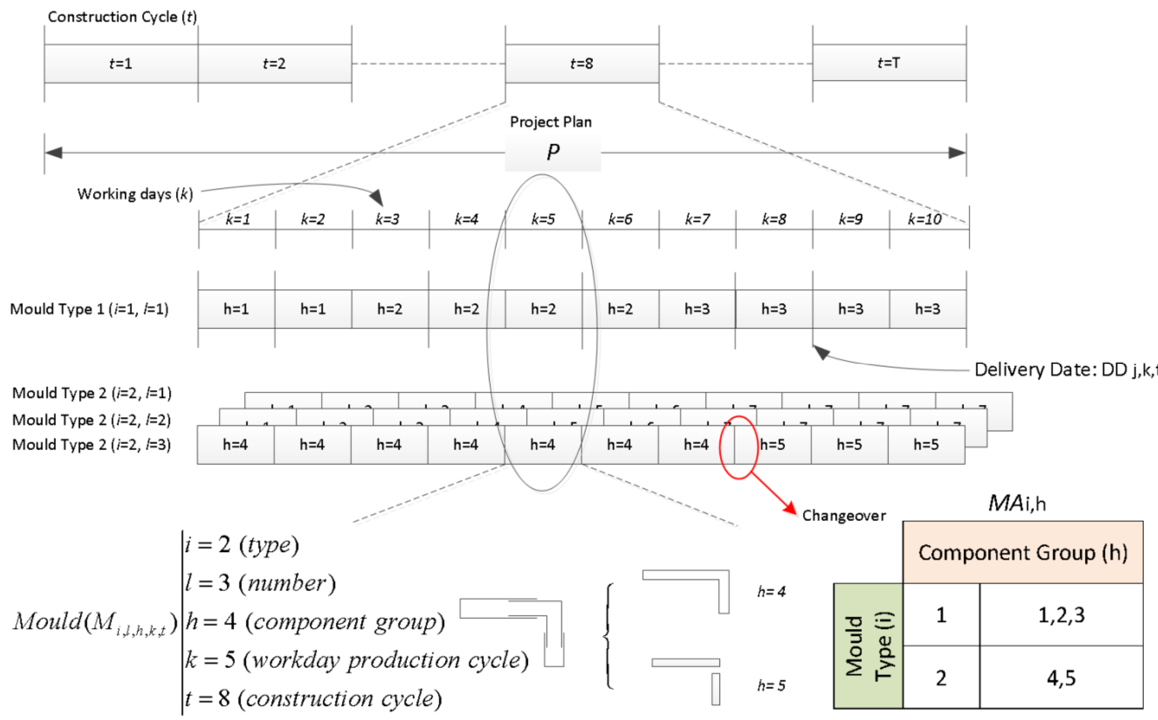


Figure 5.5: Design of production slots, construction cycle and planning horizon

A changeover is required if a mould type changes its component group from *h* to *h'* in a subsequent casting cycle ($h, h' \in H$). For example, in Figure 5.5 the third instance of mould type 2 (*i*=2, *l*=3) produces component group 4 (*h*=4) on the 7th day, but component group 5 (*h*=5) on the 8th day.

A small waste penalty is assigned to the mould if it is employed to produce a component group smaller than its capacity, e.g., if mould type 1 in Figure 5.3 is used to produce component group 3 instead of component group 1. In other words, the model encourages the utilization of a mould's full capacity.

The production of each precast component involves a known sequence of activities or production tasks. It is assumed that the productivity of moulds for producing various component groups is constant. **F** has a limited production area and stockyard (*psl*). General-purpose construction equipment (e.g., cranes, forklifts) is used for the casting and handling of elements in this factory. **F** also has a limit on its daily concrete supply (*csl*).

The objective of this study is to minimize production costs for **F** by using the minimum mould types and the minimum instances of each mould type necessary to produce all building components. The model also attempts to fully utilize each mould's capacity during its life cycle to reduce resource costs by increasing the degree of standardization. An optimal plan is achieved to satisfy installation demands for prefabricated components by minimizing mould changeovers. A deterministic scenario is used for the proposed model.

5.4. MILP formulation

The following describes the objective function and constraints in MILP. Symbols are denoted in Table 5.2.

Table 5.2: Parameters and variables symbols

Indices			
Symbol	Explanation	Symbol	Explanation
i	Index of mould type	l	Index of mould instance
j	Index of component type	h	Index of component group
k	Index of production cycle (workdays)	t	Index of construction cycle
I	Number of considered mould types	J	Number of component types
nl	Number of considered mould instance	K	Number of workdays in a construction cycle
T	Number of construction cycles	$bigM$	A big Number
Mould Parameters			
$MA_{i,h}$	Ability of mould i to produce component group h	LCi	Mould operational life
sm_i	Required workspace of mould I (m^2)	w_h	Mould Idle index for component group h
Component Parameters			
$CTG_{h,j}$	Number of component type j in component group h	vc_j	Required concrete volume for component type j (m3)
Production Plant Parameters			
$D_{j,k,t}$	Required components of type j on day k of construction cycle t	psl	Production space limit (m2)
csl	Concrete supply limit (m3)	$Wastecosth$	Penalty cost for idle mould for component group h
CH_h	Changeover cost of component group h		
Cost Parameters			
$Mouldcost_i$	Fabrication cost of mould type i		
Variables			
TPC	Total production cost	IMC	Total mould initial cost
$MchC$	Total mould changeover cost	MwC	Total mould waste penalty cost
MUC	Total mould utilization cost	$M_{i,l,h,k,t}$	Binary variable to decide Mould type i, number l, producing component group h, on day k of the t construction cycle
$Z_{i,l}$	Binary variable for adoption of mould type i, number l	$Y_{i,l,h,h',k+1,t}$	Binary variable to show mould type i, number l, producing component group h, on day k of the t construction cycle precedes to produce component group h' on day k+1

5.4.1. Mould Allocation:

In order to allocate a mould to produce a component group, a binary variable ($M_{i,l,h,k,t}$) must be defined: $M_{i,l,h,k,t}$ denotes the allocation of the l^{th} mould of type i to produce component group h on day k of construction cycle t .

$$M_{i,l,h,k,t} = \begin{cases} 1 & \text{If Mould number } l \text{ of type } i \text{ produces} \\ & \text{component group } h \text{ on day } k \text{ of } t^{th} \text{ construction cycle} \\ 0 & \text{Otherwise} \end{cases} \quad i \in I, 1 \le l \le nl, 1 \le k \le K, 1 \le t \le T$$

Each mould can produce only one component group each day. The production process takes one day and includes cleaning, casting, curing and removal. The following constraint limits the daily production of moulds:

$$\sum_{h \in H} M_{i,l,h,k,t} \le 1 \quad i \in I, 1 \le l \le nl, 1 \le k \le K, 1 \le t \le T \tag{5.1}$$

However, these moulds cannot produce a component group unless they are adopted. In order to determine the minimum number of moulds required, the following binary variable ($Z_{i,l}$) and constraint must be defined. By using $Z_{i,l}$, the computation time is significantly reduced due to the reduction in the size of the domain of feasible solutions from ($i \times l \times h \times k \times t$) to ($h \times k \times t$).

$$Z_{i,l} = \begin{cases} 1 & \text{If Mould number } l \text{ of type } i \text{ is adopted} \\ 0 & \text{Otherwise} \end{cases} \quad i \in I, 1 \le l \le nl$$

$$Z_{i,l} \times bigM \ge \sum_{h=1}^{H} \sum_{k=1}^{K} \sum_{t=1}^{T} M_{i,l,h,k,t} \quad i \in I, 1 \le l \le nl \tag{5.2}$$

As explained earlier, moulds are able to produce certain component groups. This is specified by the adaptability matrix ($MA_{i,h}$). The following equation ensures that an appropriate component group is assigned to a particular mould type:

$$M_{i,l,h,k,t} \le MA_{i,h} \quad i \in I,\ 0 \le l \le nl,\ h \in H,\ 1 \le k \le K,\ 1 \le t \le T \tag{5.3}$$

5.4.2. Mould Changeover

Mould changeover is usually dependent on the sequence of component groups. Changeover of moulds is costly and time-consuming. In order to avoid it, a penalty is applied when changeover occurs. To capture the sequence of component groups that are to be produced by a certain mould instance on two subsequent days, a binary variable must be defined ($Y_{i,l,h,h',k,t}$). This variable depicts that the specific mould type and instance (i,l) which produces a certain component group (h) on day (k) precedes component group (h') on day ($k+1$) within construction cycle (t).

$$Y_{i,l,h,h',k,t} = \begin{cases} 1 & \text{if } M_{i,l,h,k,t} \text{ producing } h \text{ on day } k \text{ preceeds to } h' \text{ on day } k+1 \text{ when } h \ne h' \\ 0 & \text{otherwise} \end{cases}$$

Thus, if $h \ne h'$ then a changeover occurs. Equation (5.4) is used to determine changeover within a construction cycle.

$$\forall i,l,h,t,k = 1,\ldots,K-1,\ \ \exists h \ne h',\quad Y_{i,l,h,h',k,t} = M_{i,l,h,k,t} \times M_{i,l,h',k+1,t} \tag{5.4}$$

As can be seen, equation (5.4) is not linear. McCormick (1976) method is applied to convert non-linear equation (5.4) to the following sets of linear equations.

$$\begin{cases} Y_{i,l,h,h',k,t} \geq M_{i,l,h,k,t} + M_{i,l,h',k+1,t} - 1 \\ Y_{i,l,h,h',k,t} \leq M_{i,l,h,k,t} \\ Y_{i,l,h,h',k,t} \leq M_{i,l,h',k,t} \end{cases} \quad (5.5)$$

As the planner knows the last component group produced by a particular mould (*i,l*) before starting the next construction cycle, $M_{i,l,h,0,1}$ is fixed to appropriately reflect that. Note that $Y_{i,l,h,h,0,t}$=1, if Mould (*i,l*) continues from construction cycle *t* to (*t+1*). Equation 5.6 forces the last component group in the construction cycle to always continue.

$$Y_{i,l,h,h,0,(T+1)} = M_{i,l,h,0,(T+1)} \quad (5.6)$$

5.4.3. Production Capacity

The daily concrete supply and production space in the factory are considered in the MILP formulation because they are common constraints. Equation 5.7 shows the space limitation in the precast plant in which sm_i denotes the required space for each mould type (*i*). sm_i is calculated based on the mould base area as well as the working and clearance area required for each mould type.

$$\sum_{i=1}^{I}\sum_{l=0}^{nl}\sum_{h=1}^{H} M_{i,l,h,k,t} \times sm_i \leq psl \quad 0 \leq k \leq K, 1 \leq t \leq T \quad (5.7)$$

The concrete supply in **F** is limited by the following constraint. Equation 5.8 calculates the total identical precast components produced by each mould type using a given parameter $CTG_{h,j}$. This parameter indicates the number of component types (*j*) in component group (*h*). In this equation, vc_j and *csl* represent the volume of component type (*j*) and the volume of daily concrete supply respectively.

$$\sum_{j=1}^{J}\sum_{i=1}^{I}\sum_{l=0}^{nl}\sum_{h=1}^{H} M_{i,l,h,k,t} \times CTG_{h,j} \times vc_j \le csl \quad 0 \le k \le K, 1 \le t \le T \tag{5.8}$$

5.4.4. Installation constraint

This constraint forces the factory, F, to meet the on-site schedule due dates. It is assumed that the capacity of the temporary storage area can accommodate the components in one construction cycle. Equation 5.9 utilizes resources to produce the required components to meet the demand DDj,k,t given to the planner. In Equation 5.9, DDj,k,t denotes the total number of components (j) required on the kth day of construction cycle (t). An alias α is used for k to calculate the total demands of component (j) by day (k-1).

$$\sum_{i=1}^{I}\sum_{l=0}^{nl}\sum_{\alpha=1}^{k}\sum_{h}^{H} CTG_{h,j} \times M_{i,l,h,\alpha,t} - \sum_{\alpha=1}^{k-1} DD_{j,\alpha,t} \ge DD_{j,k,t} \quad 1 \le j \le J, 1 \le k \le K, 1 \le t \le T \tag{5.9}$$

5.4.5. Planning objective function

The objective of this research is to minimize production costs. The production costs considered in this study include the initial mould fabrication cost (*IMC*), the cost of mould usage and replacement (*MUC*), the mould changeover cost (*MchC*) and the penalty cost to minimize partial utilization of moulds (*MwC*).

Objective function:

$$Minimize\ TC = IMC + MUC + MchC + MwC \tag{5.10}$$

Where:

$$IMC = \sum_{i=1}^{I} \sum_{l=1}^{nl} Z_{i,l} \times MouldCost_i \qquad (5.11)$$

$$MUC = \sum_{i=1}^{I} [\sum_{l=0}^{nl} (\frac{\sum_{h=1}^{H} \sum_{k=1}^{K} \sum_{t=1}^{T} M_{i,l,h,k,t}}{LC_i}) \times MouldCost_i] \qquad (5.12)$$

$$MchC = \sum_{i=1}^{I} \sum_{l=0}^{nl} \sum_{h=1}^{H} \sum_{h' \neq h}^{H} \sum_{k=1}^{K} \sum_{t=1}^{T} (Y_{i,l,h,h',k,t} \times CH_h) \qquad (5.13)$$

$$MwC = \sum_{h=1}^{H} [(\sum_{i=1}^{I} \sum_{l=0}^{nl} \sum_{k=1}^{K} \sum_{t=1}^{T} M_{i,l,h,k,t}) \times w_h \times WasteCost_h] \qquad (5.14)$$

5.5. Numerical Evaluation and Case Experiment

To study the performance of the proposed model, two examples and various scenarios for planning and mould optimization are considered. The first example is designed to validate the proposed optimization model while the second example is designed to evaluate the performance of the model at handling real-size projects. For model implementation,CPLEX 12.2.0.2 in GAMS 23.6.5 was used by a Dell Precision T5500 with Intel(R) Xeon (R) CPU X5650 @ 2.67 GHz Processor with 48 GB of RAM running Windows 7 Professional 64-bit operating system.

5.5.1. Example 1

Thirteen blocks of 10-storey residential buildings are selected to be constructed using the prefabrication method. The plan of the storeys is typical and it is the same as the plan given in Figure 5.1a. The construction cycle for this project is defined as the production and installation of the total required components for two storeys for each of the five blocks. Thus, the total number of construction cycles is 5. The required components for each construction cycle are produced in10 days (*K*=10). The production area for this project is limited to 500 m^2. The concrete supply and production capacity of the producer is 150 m^3/day. The mould life cycle is assumed as 100 times.

The proposed model is performed for three scenarios. The first scenario (Scenario 1) is for prefabrication configuration comprising complex components. This configuration is obtained using the CoCoPS framework mentioned earlier. In this scenario, moulds are allowed to produce the composition of components (component grouping). The second scenario (Scenario 2) analyses the same prefabrication configuration, but with dedicated moulds. In other words, component grouping is not applied. The third scenario (Scenario 3) is prefabrication using the traditional prefabrication method in which elemental precast components are adopted. This scenario is designed to evaluate the cost advantage of employing a higher degree of prefabrication (as in Scenarios 1 & 2).

The list of building components required for scenarios 1 and 3 is shown in Table 5.3 together with the quantity of each component type for one storey. The volume of components is also presented.

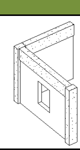
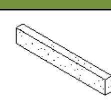
Table 5.4 shows the *CTG* matrix for the configuration of Scenarios 1 and 2 derived from the CoCoPS framework. Although more component groups are possible with this configuration of components,

Table 5.4 presents only eighteen groups as examples for this case study. The initial number of mould types considered is equal to the number of component types (I=J=5). In order to bound the search domain, the maximum number of each mould type (nl) is set at 10. Mould information including the required space, fabrication cost and changeover cost is presented in Table 5.5. The ability of moulds to produce different component groups (mould adaptability matrix) is shown in

Table 5.6. The waste index (W_h), which is computed based on the unused volume of moulds for the production of different component groups, is also presented in this table for each of the component groups.

The installation sequence and delivery date of the produced components for one construction cycle is presented in Table 5.7 so that on day 4, for example, 15 components of type 1, 6 of type 2, 2 of type 4 and 2 of type 5 are required.

Table 5.3: Component information for each configuration

Scenario 1				Scenario 2			
Component Type	Shape	Qty	Volume	Component Type	Shape	Qty	Volume
1		2	5.48	1		5	.72

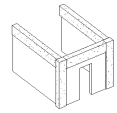
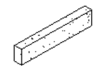
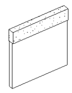
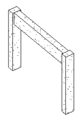
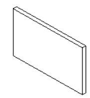
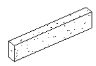
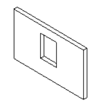
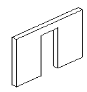
2		1	9.96	2		5	.36
3		1	2.28	3		8	.4
4		1	1.52	4		5	1.92
5		1	.36	5		2	1.7
				6		1	1.13

Table 5.4: Given component groups and mould waste information

Component Type *j*	Component Group *h*																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	1	3	3	2	0	0	1	1	0
4	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	2
5	0	0	1	0	2	0	0	0	1	0	0	0	1	3	1	1	2	0

Table 5.5: Mould properties and mould cost information

Mould Type	Concrete Volume (vc_j) (m^3)	Mould Space (sm_i) (m^2)	Mould Fabrication Cost ($Mouldcost_i$) (\$)	Changeover Cost (CH_i) (\$)
1	7.2	18	25,000	200
2	3.8	9	21,000	150
3	1	2	12,000	100
4	0.8	1.5	9,000	100

Table 5.6: Mould adaptability matrix (MA) and waste penalty index

Mould Type i	Component Groups																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Waste Index (W_h)	0	0.8	1.6	1.6	3.8	2.1	0	0.8	1.2	0	0.4	0.8	2.1	3.8	1.6	0.4	0	2.4

Table 5.7: Demand profile for one construction cycle for Example 1

Component Type j	Demand (D) on day (k)										
	1	2	3	4	5	6	7	8	9	10	*Total*
1	-	-	-	15	5	1	4	-	10	5	40
2	-	-	-	6	4	-	-	-	-	10	20
3	-	2	-	-	-	-	10	-	-	8	20
4	3	-	4	2	2	1	-	5	1	2	20
5	-	4	-	2	4	-	4	-	4	2	20

The solution statistics for the three scenarios are shown in Table 5.8. The average solving time of the example project for all steps of the recursive procedure is about 38 seconds. The relative GAP is less than 1%. This reflects the difference between the obtained objective function and the lower bound in which all variables are relaxed to take any value from 0 to 1. The optimal solution for Scenario 1 is achieved with a total project cost *TC*= \$394,200 in which *IMC* is \$253,000, *MchC* at is \$1,600, *MwC* at is \$9,600 and

MUC is$130,000. The production cost for the traditional prefabrication method (Scenario 3) is 13.27% higher, which shows that adopting a higher degree of prefabrication (using complex moulds and component grouping) can significantly reduce production cost. The total project cost for Scenario 3 is *TC*=$446,500 in which*IMC*is$299,000, *MUC*is$147,500 and *MchC* and *MwC*are0. Since production is based on the prefabrication of individual building elements, each mould is allowed to produce one dedicated building element so that there is no changeover cost and mould waste penalty. The waste and changeover penalties force the model to have only changeovers during the construction time for the first scenario and to fully utilize complex moulds to produce combinations of smaller component types. Total time to solve scenario 1, 2, and 3 are 425.35, 0.36, and 326 seconds respectively. Total computational for scenario 2 is very low because the concept of grouping has not been considered.

As can be seen in Table 5.9, only two mould types (i=2 and i=5) were adopted to produce all of the required component types in Scenario 1. This table also shows that the manufacturer needed 8 moulds comprising of 6 of mould type 2 and 2 of mould type 5. In other words, mould types 1, 3 and 4 were not needed in this project. However, in Scenario 3, the manufacturer must employ 54 moulds to produce all of the required building elements on time. Further to the increase in mould fabrication and mould usage costs, adopting more moulds requires more space and workers in the plant which would increase indirect costs and reduce the capacity to handle multiple projects at the same time.

Table 5.8: Model and Solution statistics for Example 1

model statistics	Example 1

	Scenario 1	Scenario 2	Scenario 3
Binary Variables	941,500	72,801	307,251
Constraints	2,270,551	169,151	717,251
Nonzeros	6,814,951	596,351	2,441,101
Gap (%)	0.0	0.0	1.0
MILP objective ($)	394,200	494,800	446,500
Mould Initial Cost	253,000	311,100	299,000
Mould Usage Cost	130,000	183,700	147,500
Waste Penalty Cost	9,600	0	0
Changeovers Cost	1600	0	0

Table 5.9: Adopted mould to produce all component types for Example 1

Mould Type *i*	Mould Number (*l*)					
	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	1	1	1	1	1
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	1	1	0	0	0	0

The schedule of production can be extracted from the mould variable ($M_{i,l,h,k,t}$). The production schedule of the case study for all three scenarios is presented in Figure 5.6-5.8 as a Gantt chart. The horizontal axis shows time (workdays) and the vertical axis shows the moulds adopted to produce the components. The horizontal rectangular bar denotes the component group that is produced with the employed mould at the corresponding time interval (indicated as a label). As can be seen in Figure 5.6, a complex mould type (i=2) is utilized to produce three different component groups (7, 8 and 10). However, there are only 4 changeovers in production for one construction cycle. As a result, component grouping facilitates the model to produce a combination of simple components using complex moulds. Moreover, there is no mould idle time. As can be seen in Figure 5.8, besides the increased space and crew required for production in scenario 3, the idle time of the moulds is 17 mould-days per construction cycle.

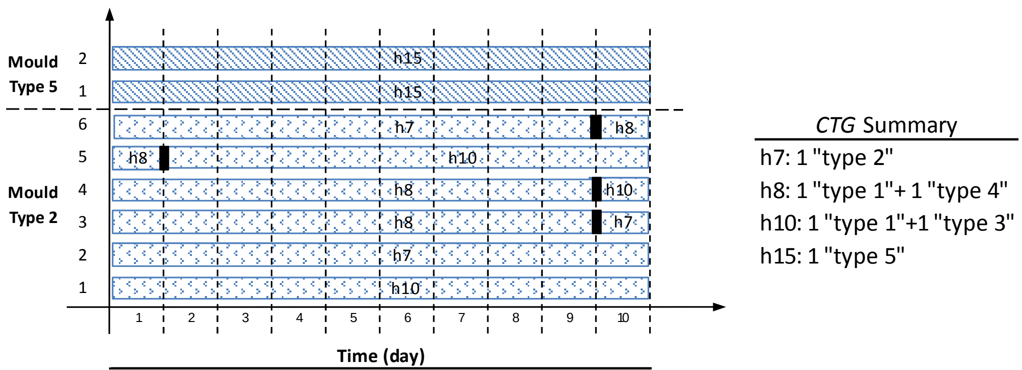


Figure 5.6: Optimized production plan of Example 1 for Scenario 1

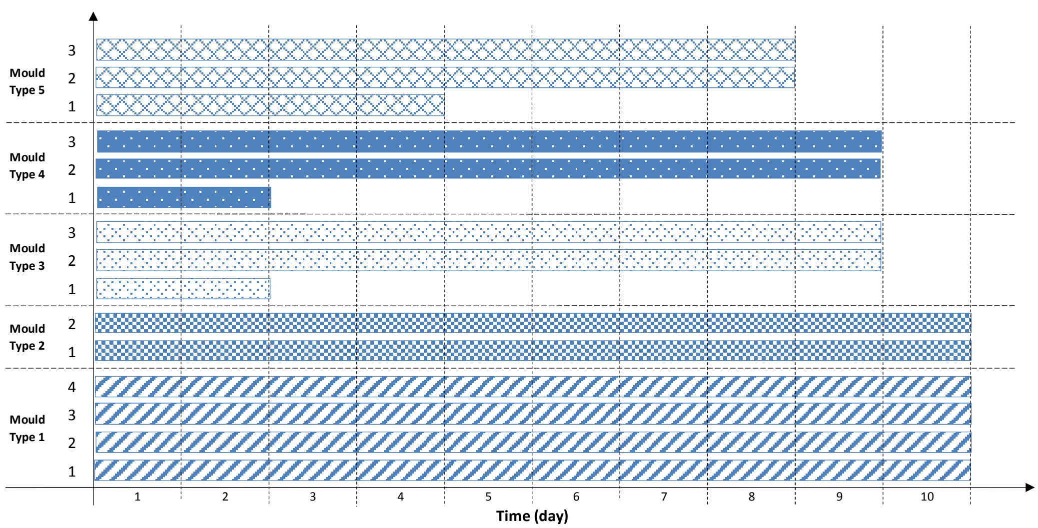


Figure 5.7: Optimized production plan of Example 1 for Scenario 2

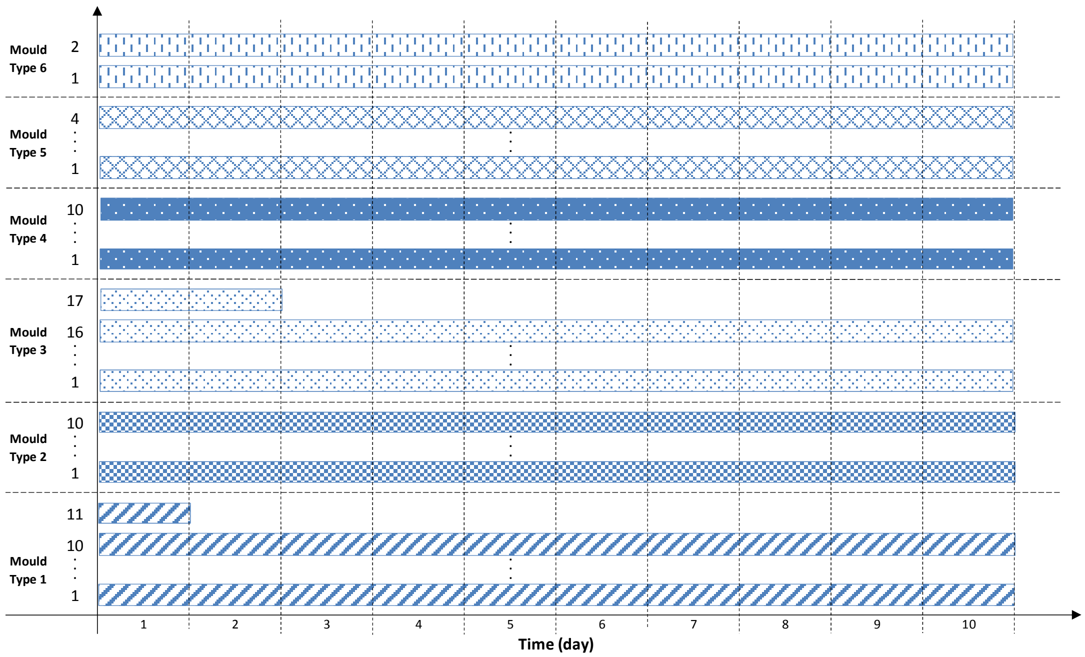


Figure 5.8: Optimized production plan of Example 1 for Scenario 3

In Scenario 2 where dedicated moulds are adopted for a higher degree of prefabrication configuration, the total project cost is *TC*= *IMC*+*MUC*=$494,800. Although the mould changeover cost and mould waste penalty are 0, the total production cost of Scenario 2 is 25.52% higher than the optimal solution (Scenario 1) and 10.82% higher than the traditional prefabrication method.

The *IMC* and *MUC* of Scenario 2 show that if the component grouping concept is not adopted for using more complex moulds, the mould utilization will be lower. Furthermore, complex moulds occupy more space in the plants so there may be space constraints for multiple projects. The mould idle time increases to 30 mould-days per construction cycle (Figure 5.7).

In the end, a higher degree of prefabrication without component grouping is even more expensive than the traditional prefabrication method. The component grouping concept provides a mechanism for more effective utilization of complex moulds so that the advantage of a higher degree of prefabrication can be realized.

5.5.2. Example 2

The second example highlights the ability of the proposed model in handling real-sized problems. Moreover, this example demonstrates that the economic advantage of adopting a higher degree of prefabrication in real projects is still significant.

In this example, a medium-sized residential building comprising 20 storeys is selected to be constructed using prefabrication method. A construction cycle in this example is defined as prefabrication of required components for two storeys so that the number of construction cycles is 10 (T=10). The production area for this project is limited to 1500 m^2. The concrete supply and production capacity of the producer is 200 m^3/day. Each storey consists of 159 beams, 143 columns, and 89 walls in various sizes. These components should be produced during the 10 days for each construction cycle (K=10). The *CTG* matrix obtained from CoCoPS framework indicates that there are 58 component groups for this configuration. The model execution is repeated for the same three scenarios in Example 1.

The solution statistics for the three scenarios are shown in Table 5.10. The average solving time for all steps of the recursive procedure is about 22 minutes. The relative GAP in the worst scenario is 1.6%. However, a lower GAP could be achieved with a higher computation time.

The optimal solution is achieved for Scenario 1 with the total project cost *TC*= \$907,200 in which *IMC* is at \$659,000, *MchC* at \$8,400, *MwC* at \$31,400, and *MUC* at \$208,400. Nevertheless, the total production cost using the traditional prefabrication method (Scenario 3) is \$990,400 which is 9.17% higher than Scenario 1. This analysis also indicates that, although, more complex moulds may be required for adopting a higher degree of prefabrication, a lower number of moulds and higher utilization of complex moulds (through component grouping) significantly reduces the total production cost.

The total saving is slightly less than earlier example possibly because of the different mix of component types. Further to this, the number of some of the building elements is divisible to the construction cycle so that mould utilization can be increased by these elements.

In this example, 25 moulds from 5 types are required to produce all components. The total idle time of moulds is only 8 days in each construction cycle. In contrast, using Scenario 3, the manufacturer requires 85 moulds altogether to meet the schedule demand. Moreover, mould idle time increases to 105 days in comparison with Scenario 1.

Table 5.10: Model and solution statistics of Example 2

model statistics	Example 1		
	Scenario 1	Scenario 2	Scenario 3
Binary Variables	17,057,900	4,385,203	123,934,805
Constraints	41,672,901	8,642,766	345,723,702
Non-zeros	116,103,001	23,540,971	899,743,231
Gap (%)	0.3	0.8	1.6
MILP objective ($)	817,200	902,500	890,400
Mould Initial Cost	659,000	790,800	757,800
Mould Usage Cost	118,400	126,700	132,600
Mould Waste Cost	31,400	0	0
Changeovers Cost	8,400	0	0

In Scenario 2 where dedicated moulds are adopted, without component grouping, total project cost increases by 23.8% compared to Scenario 1 ($TC = IMC + MUC =$ $1,123,100) and similar to Example 1, the total cost of Scenario 2 is 13.4% higher than traditional prefabrication method as well. The second scenario in both examples highlights the significance of component grouping and mould adaptability in the adoption of a higher level of prefabrication.

The second example highlights that in a real project, where more complex components are required, the idea of adopting a higher degree of prefabrication using prefabrication configuration and component grouping may reduce the project cost up to 9%. Moreover, further cost savings is possible due to lower space, equipment, and workers for each project because of less number of moulds utilized.

5.6. Concluding Remarks

In order to evaluate the prefab configurations generated by CoCoPS framework to employ a higher degree of prefabrication, this chapter integrates two forms of industrialization, namely prefabrication and standardization, to propose a solution for precast project planning from design and production perspectives. The proposed approach for mould optimization in precast production may give designers more flexibility to meet new market demands. Moreover, the proposed model has the potential to significantly reduce production costs in the precast industry through optimal mould allocation and lean operation

To employ a higher degree of prefabrication, two new parameters were introduced through two matrices: Component Type per Group (*CTG*) and Mould Adaptability (*MA*). The *CTG* presents the composition of component types that can be produced by each particular mould, while the *MA* describes the capability of moulds to produce component groups.

The proposed MILP model accommodates and presents a treatment for key aspects of precast production planning resources and activities such as mould allocation, sequence-dependent changeovers, delivery dates, etc. and gives the exact number of required moulds during the planning horizon and a schedule for each mould.

The model was validated through two case studies. The results demonstrate that adopting a higher level of prefabrication can significantly reduce resource costs by employing more complex moulds for the production of various component types. The proposed model can reduce production costs by over 9% compared to the traditional elemental approach.

The case study also shows that the adoption of complex moulds must be accompanied by high utilization of the moulds. This is achieved through the novel concepts of mould adaptability and component grouping. Without incorporating these concepts, the use of complex moulds (Scenario 2 in the case examples) actually results in production costs more than 10% higher than the traditional elemental approach.

In addition to the lower initial mould fabrication and mould utilization costs, a higher degree of prefabrication has other potential benefits such as a reduction of indirect costs through a smaller production area and fewer workers. For a project as a whole, a higher degree of prefabrication also leads to faster erection cycles. A future research line in prefabrication planning could expand the model components across projects to achieve a higher degree of standardization throughout the factory. Furthermore, other cost constituents and detail constraints, such as transportation within the factory, could be added to the objective function.

CHAPTER 6: CONTAINER LOADING OPTIMIZATION FOR TRANSPORTATION OF PREFABRICATED COMPONENTS

In this chapter, the second optimization model to obtain the best generated configuration of precast elements is studied. Transportation is identified as the second expensive resource in off-site projects. The adoption of higher degree of prefabrication comprises irregular and complex components (rather than cuboid individual elements) may increase the transportation cost. To ensure that the trucks are using their maximum capacity for transportation of produced components from factory to construction site, an optimization model is proposed. The proposed model attempts to reduce the transportation cost of produced component by packing these components into minimum number of containers.

This problem is a particular form of a general Container Loading Problem which is NP-complete problem. A new approach is proposed to combine a heuristic sliding algorithm and GA sequence generator. The sliding algorithm developed in this study uses voxels to represent non-standard irregular 3D objects and place them into the containers. The sliding algorithm can accommodate any orientation and size restrictions using voxel representation. A genetic algorithm is used to generate sequences of the input objects to be allocated. The evaluation criterion of loading is determined based on loading strategies. The performance of the proposed model is evaluated through a case study with various scenarios.

6.1. Introduction

The previous chapters described the development of the framework in which individual prefabricated elements are combined to configure a higher degree of prefabrication. In order to find the best configuration of elements in terms of total cost, two optimization models are proposed. The first optimization model attempts to minimize the production cost by maximizing the mould utilization, while, the second optimization model is designed to minimize the transportation cost.

The transportation issue can be studied from two perspectives. A common study perspective is the total cost optimization of transportation between factory, warehouse and construction site. This perspective has been studied by many researchers such as (Chan and Lu (2005), Huang et al. (2005), K.C. Shih (2005)). However, another perspective is to minimize the transportation cost through containerization. This perspective has been addressed as a valuable area to be explored. Since a higher degree of prefabrication employs flexible design and complex prefabricated components, the complexity of containerization is increased. Thus the transportation issue in this research is defined as minimizing the total number of trucks to be loaded for transportation of produced component to the construction site.

This problem is addressed as a particular form of general NP-hard Container Loading Problem (CLP). CLP is an extension of the classical Bin-Packing Problem (BPP) and consists of packing a given set of boxes into a chosen set of containers with variable sizes and resulting in minimum shipping costs. Either one of the referred problems is mathematically very challenging since, in practice, the solutions need not only implement a good usage of the space available but also meet all the practical constraints involved.

This chapter provides performance enhancement to container loading in the precast industry by proposing a novel representation method and an emerging algorithm for CLP. The problem is to find out the best sequence of locating 3D irregular-shaped objects that are not restricted to any shapes or orientations into 3D rectangular containers. The stability of components and loading capacity (weight) are the problem constraints. An automated system is developed to combine a heuristic sliding algorithm and GA sequence generator. The sliding algorithm developed in this study uses voxels to represent non-standard irregular 3D objects and place them into the containers. The sliding algorithm can accommodate any orientation and size restrictions using voxel representation. A genetic algorithm is used to generate sequences of the given objects to be allocated. The evaluation criterion of loading is the unused space percentage of each allocation.

6.2. Container Loading Problem Model

Bin packing problem has been well-researched for more than 40 years, as it is one of the fundamental and challenging problems in the area of optimization. The BPP or CLP are NP-complete (Garey and Johnson, 1990). The problem becomes more complicated for variable length container, sequence-based problems or irregular shapes. Because of the diversity of structures in real world BPP/CLP problems, there exists no general standard method for solving them.

Several methods have been used to solve CLP. Depending on the problem requirements, the techniques may attempt to minimize number of containers, minimize wasted space, maximize profit, or stabilize the balance of containers or even combination of these objectives. Being a combinatorial problem, 3D bin packing is usually solved using either optimization, heuristic or emerging algorithms. Optimization algorithms try to deliver an optimal and precise solution; however, heuristic algorithms provide an acceptable solution in a relatively reasonable time (which is linear time with respect to the input size). The emerging solutions try to combine combinatorial and heuristic methods to take the advantages and overcome the disadvantages of each of them.

These approaches are summarized in many review papers (Bortfeldt and Wascher (2012), Bischoff and Ratcliff (1995), Egeblad et al. (2010)). Most of the research focuses on the orthogonal placement of rectangular objects into rectangular containers, with no additional constraints. However, in real practice, industries such as prefabrication, steel manufacturers and shipbuilding have non-standards 3D objects with some constraints (e.g. order of delivery, stability of truck, weight distribution). Some of these constraints are addressed in literature review as well (Egeblad et al., 2010; Fasano, 2004).

Non-standard packing problems with additional constraints are often studied using Mixed Integer Programming (MIP) (Schepers, 2004), or heuristics (Fasano, 2008). The MIP models have restrictions in application due to the NP-complete nature of the problem. Most of the applications consist of a very large number of variables, so the solution time exponentially increases by the problem size and may not be practical. Furthermore, moving from an optimal fractional-valued solution to an optimal integer-valued solution is not straightforward. Defining irregular-shape objects increases the difficulty of using deterministic methods. In order to overcome these difficulties various heuristic approaches are proposed.

Several models have been developed using heuristic and meta-heuristic methods such as GA, Tabu Search (TS), Simulated Annealing (SA), and tree search. These models are based on different heuristic packing approaches, such as the guillotine-cutting, stack-building, wall-building, or block-building approach (i.e., the cuboid arrangement approach). An extensive literature review on these approaches has been conducted by Pisinger (2002).

For example, in wall building approach the container is loaded by objects along the longest side of the container in vertical layers (Walls) (Pisinger, 2002). Although these studies can find optimal (or nearly optimal) solution for homogeneous and heterogeneous cubic objects, irregular-shaped objects are rarely considered. This is mainly because for non-rectangular objects the geometric complexity for representation of objects and for allocation algorithms is generally prohibitive.

Further to this, the sequence of allocation in CLP generates a large set of solutions. In the context of CLP, Genetic algorithm enhances the traditional approach when the number of possible 3D objects and the sequence of objects are increased. Poshyanonda and Dagli (1992) combine the genetic algorithm with a heuristic method to solve a two-dimensional bin packing problem. They have used genetic algorithm to generate a proper sequence of the required configuration to the heuristic allocation algorithm by exploring the search space.

Instead of using only deterministic or heuristic approaches as a solution to BPP/CLP, many researchers have been investigating the possibility of combining both methods into a solution approach to take benefits and overcome the disadvantages of each of them. An example of this type of solution is proposed by Lin et al. (1991). In their model, a heuristic expert system generates many possible solutions using knowledge bases that contain the rules related to loading restrictions, unloading sequence, nesting methods, and allocation goals. Then, a linear programming model is used to achieve those solutions which will result in a minimum number of containers.

In order to overcome the challenges of component representation, sequence of components, and formulating constraints, an emerging approach is developed in this study. This approach utilizes a new representation method to simplify and abstract the complex geometric equations to three-dimensional volumetric pixels. The voxel presentation is defined by a three dimensional binary matrix that significantly reduces the computational time. A new Container Loading Algorithm (CLA) is developed to slide 3D objects into the containers considering their best fit angles. The CLA is followed by a novel reuse algorithm to obtain a higher loading density. Genetic Algorithm is used to obtain the optimal sequence of components for CLA.

Although this chapter aims to optimize the transportation of precast components to the construction site, the method can be generalized to any container loading problem. A solution approach that combines these emerging techniques to solve a general three-dimensional container loading for irregular-shaped objects is described in the following sections.

6.3. Problem Statement

The problem is locating a certain number of 3D objects (*N*) that are not restricted to any particular shape or orientation into minimum number of trucks. Several packing strategy for placement of 3D objects into trucks can be considered (e.g. fully packed or constant distribution). Holes within an object are considered to be a free space to be utilized for loading another object. A margin is considered between objects that may be required for loading and unloading. Moreover, this margin is utilized for fixing and stabilizing the objects in the container.

Generally, the container is defined as 3D cuboid which can be divided into two different types, namely, containers which are restricted in all width, length, and height (e.g. real truck) and containers restricted only in width and height having infinite length (e.g. stock yard). Although the real truck situation reflects the issue of transportation in this study, the container with infinite length is also considered in case study to show the performance of the proposed model to obtain a high loading density.

The objective of the problem is to maximize the container space usage according to packing strategy or minimize transportation cost function, $f(N)$, which is in this case, the minimum number of containers required to allocate all 3D objects. The problem of placing a set of N objects (prefabricated components) ($C_1, \ldots, C_N$) into container(s) can be defined as:

Minimize $f(N)$ = Total cost of hiring trucks to transport N objects

$$= CT.T + PSP \tag{6.1}$$

In which *CT* is cost of hiring a tuck and *T* is the number of trucks that are required to be used. *PSP* is Packing Strategy Penalty that is applied according to the desired packing strategy. In this study, the produced components are heavy and bulky so that the ease of access for loading and unloading must be considered in the packing strategy. Therefore, a constant distribution is selected for the packing strategy to avoid highly packed trucks. Therefore in order to keep the loading density of the trucks about the average of the loading density *PSP* is defined as $CP.\sum_{t=1}^{T}(\overline{LD}-LD_t)^2$ to achieve the constant packing strategy. The term *CP* in the objective function is the penalty cost for the packing strategy, $\overline{LD}$ is the average loading density, and LD_t is the loading density of truck *t*. The model is subjected to the following constraints:

- Component must be totally placed within the container's boundary:

$$\begin{cases} z_{ov} \geq 0 \\ z_{ov} \leq H \\ y_{ov} \geq 0 \\ y_{ov} \leq W \\ x_{ov} \geq 0 \\ x_{ov} \leq L \end{cases} \tag{6.2}$$

Where *W*, *H* and *L* are the width, height and length of the truck(s) respectively and (x_{ov}, y_{ov}, z_{ov}) is the location of the v^{th} vertices that represents the component (C_o).

- There must be no overlaps between Components (C)

$$\forall o, o' \; \exists \; o \neq o', \; C_O \text{ does not overlap with } C_{O'} \tag{6.3}$$

The overlap check is done by adding the object voxel to the container current situation voxel. If the result is more than "1" the overlap has occurred.

- Component must be stable within the container

$$\forall C_o \in \{C_1, ..., C_N\} \ \mathrm{C}_o \text{ must be stable in the truck} \qquad (6.4)$$

Vertical and horizontal stability of components are checked in this study. Vertical stability deals with the situation when the container is not moving. It prevents components from falling down onto the container floor or on top of other components. Several approaches have been addressed in literature for vertical stability checking. According to the nature of objects (prefabricated components) in this study, direct and indirect support of center of gravity reported by (Mack et al., 2004) is selected for our model.

Horizontal stability assures that components are not shifted significantly when the truck is moving. For horizontal stabilizing of components within a truck the following criteria which is termed as "interlocking" introduced by Carpenter and Dowsland (1985) and Bischoff (1991) are used:

- The base of each component must be in direct contact with the top surface of truck or underneath components.
- At least 80% of the base are of each component must be supported by below components.

- The sum of the weights of loaded components must be smaller than or equal to the weight limit imposed by the container.

$$\forall\ t\ ,\ \sum_{o=1}^{O_t} W_{C_o} \leq TLL_t \qquad (6.5)$$

Where t is index of truck, o is index of object, O_t is number of components in a particular truck (t), W_{C_o} is weight of component C_o , and TLL_t is loading limit of truck (t).

To evaluate the performance of different obtained results, loading density of the container is used as a comparison unit which is defined as:

Loading density (L) = Total usage space / Total available space (in a truck)

6.4. System Overview

The system developed in this research for container loading consists of three main modules: preprocessor module, Container Loading Algorithm (CLA) module, and Sequence Generator module. The proposed system is illustrated in Figure 6.1.

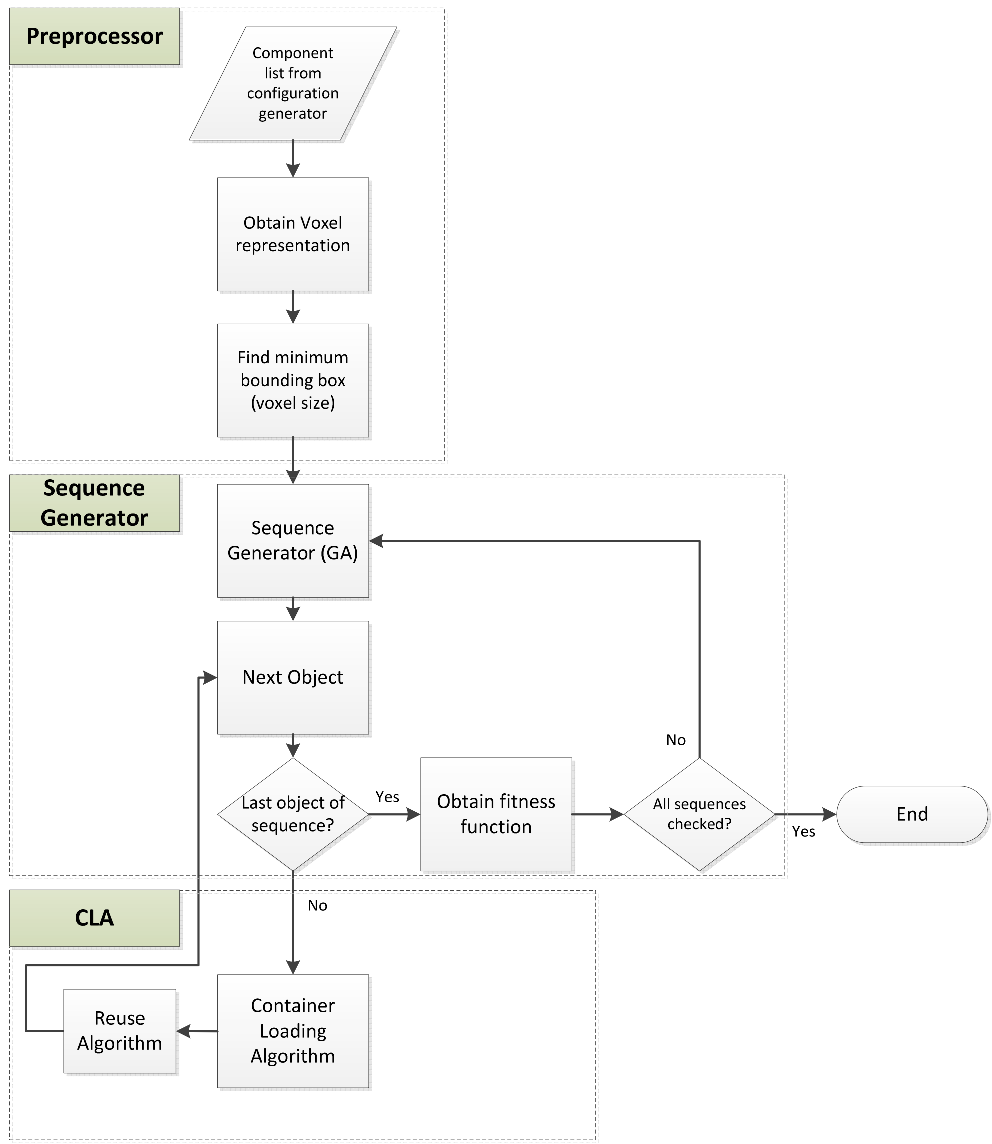


Figure 6.1: Proposed system for locating prefabricated components into trucks

6.4.1. Preprocessor

Essentially, the preprocessor parses the input data acquired from the Configuration Generator and then provides an appropriate representation of building components to its receptors (Sequence Generator and Container Loading Algorithm (CLA)). The preprocessor consists of two key functions. The first function converts boundary representation (Brep) to Voxel representation. The second function is a simple heuristic to orient the objects so that they require minimum bounding box (voxel size) for their individual placements in trucks.

6.4.1.1. Representation schema

The Boundary Representation (Brep) has been employed in CoCoPS framework for 3D object representation. In Brep, each object is represented by a list of vertex coordinates $[(x_1, y_1, z_1), \ldots, (x_M, y_M, z_M)]$ where M is the number of vertices used to represent the object. During the placement process, the overlap among the objects in the container should be identified. With coordinate representation, the overlap test is done by checking intersections between segments' line of the objects located on the container. When the number of objects located on the container is increased, the process time increases exponentially. Moreover this process is more challenging when the model is handling three-dimensional irregular-based shapes. In order to overcome this difficulty, the volumetric pixel (voxel) representation is proposed. Each object (O) is represented by a voxel or 3D matrix $V_o[X_o,Y_o,Z_o]$. Where, X_o,Y_o,Z_o are the length, width, and height of the smallest cubic enclosure of the object. The size of $V_o[X_o,Y_o,Z_o]$ is determined as follows:

$$X_o = \lceil OL_o / PF \rceil$$

$$Y_o = \lceil OW_o / PF \rceil$$

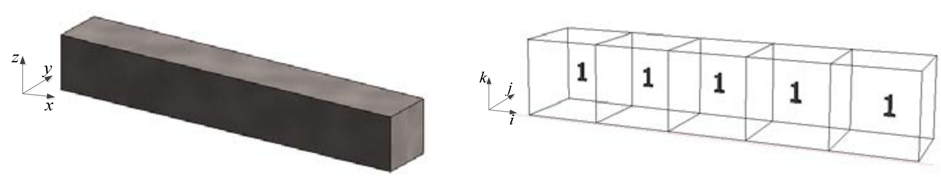
$$Z_o = \lceil OH_o / PF \rceil \qquad (6.6)$$

Where, *OL*, *OW* and *OH* are the dimensions of the bounding box of the object. *PF* is a preferred precision factor by which the computational accuracy is determined. The lower values of the precision factor lead to higher computation time.

The value of the voxel in its local coordinate system (*i, j, k*) is defined as:

$$V_{i,j,k} = \begin{cases} 1 & \text{location (i,j,k) is filled by object} \\ 0 & \text{otherwise} \end{cases} \qquad (6.7)$$

For example, in Figure 6.2a a straight rectangular beam is shown with dimensions of 100cm x 20cm x 20cm. If the precision factor is 20 cm, the size of voxel (Vo) for this beam is [5,1,1]. Another example is an irregular-shaped building component and its schematic voxel representation which is shown in Figure 6.2b



(a) (b)

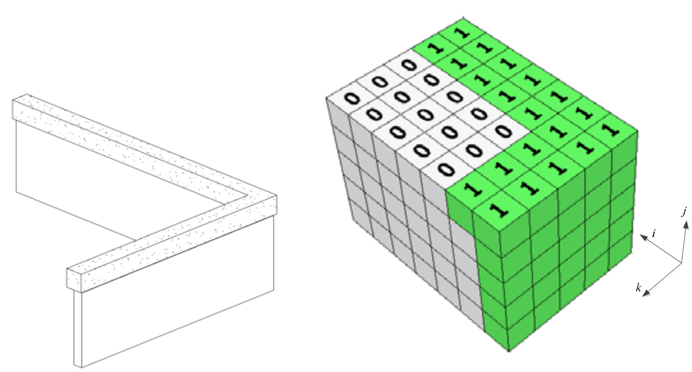


Figure 6.2: (a) Sample voxel representation of a straight beam V[5,1,1] (b) Sample voxel representation of L-Shaped building component V[7 ,5 ,5]

A truck or container is also represented by the voxel T(c) where $T_{i'j'k'}(c)$ is the value in the global coordinate system (*i', j', k'*) on the truck after the first "c" components have been placed. $T_{i'j'k'}(c)$ is computed as shown in Equation 6.8:

$$T_{i'j'k'}(c) = T_{i'j'k'}(c-1) + (R \times V_{ijk}(c) + T) \quad \text{and} \quad \mathrm{T}_{i'j'k'}(0) = 0 \tag{6.8}$$

Where $T_{i'j'k'}(c-1)$ is the value of the voxel in the global coordinate (*i', j', k'*) on the truck after the first (*c-1*) objects have been placed. $T_{i'j'k'}(0)$ is the value of the voxel on the truck at the beginning of the placement process. $V_{ijk}(c)$ is the voxel representation of component (c) at its local coordinate system (*i, j, k*) to be placed into the truck. The *R* and *T* are rotation and translation matrices to convert the local object voxel coordinate system to the truck global coordinate system respectively. The detail of this conversion was described in Equations 4.2 and 4.3.

The overlap of components in the truck is detected by a value greater than one in the truck voxel. Although the overlap test can be performed easily and quickly, there is a need for a large memory for this representation schema. To this end, binary voxel is employed rather than the integer. Consequently, a flag is assigned to overlap cells when detected.

6.4.1.2. Finding Minimum Bounding Box (MBB):

Different orientations of an object may result in different free spaces in the container. A bounding box which surrounds the whole object is used as an approximation of the volume required to place the object into a container. It is required to obtain the smallest voxel size of the bounding box. An object with different orientations may have different sizes of bounding box. In this regard, a reference point is required for placement and rotation of components. In this study, the center of gravity (G_x, G_y, G_z) is used as the reference point for each object. It is calculated using the corresponding voxel representation. The following formula is utilized to calculate center of gravity assuming uniform density for building components.

$$G_x = \frac{\sum_{i=1}^{X}(\sum_{j=1}^{Y}\sum_{k=1}^{Z} C_{ijk}).i}{\sum_{i=1}^{X}\sum_{j=1}^{Y}\sum_{k=1}^{Z} C_{ijk}} \quad G_y = \frac{\sum_{j=1}^{Y}(\sum_{i=1}^{X}\sum_{k=1}^{Z} C_{ijk}).j}{\sum_{i=1}^{X}\sum_{j=1}^{Y}\sum_{k=1}^{Z} C_{ijk}} \quad G_z = \frac{\sum_{k=1}^{Z}(\sum_{i=1}^{X}\sum_{j=1}^{Y} C_{ijk}).k}{\sum_{i=1}^{X}\sum_{j=1}^{Y}\sum_{k=1}^{Z} C_{ijk}} \tag{6.9}$$

In order to find the smallest bounding box of an object, 90 different orientations (α =0,1, …,89) around each of three axes (x,y,z) of the object are obtained and the angle α_{min}=(α_x, α_y , α_z) that generates the smallest voxel size is selected as the orientation of the object for placement in the container. All calculation is done in local coordinate system. The local coordinate system is converted to the global system during the CLA implementation. Rotation matrices around different axes are as follows:

$$R_x(\alpha_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_x) & -\sin(\alpha_x) \\ 0 & \sin(\alpha_x) & \cos(\alpha_x) \end{bmatrix}, \text{ where } \alpha_x \in (0,...,89)$$

$$R_y(\alpha_y) = \begin{bmatrix} \cos(\alpha_y) & 0 & \sin(\alpha_y) \\ 0 & 1 & 0 \\ -\sin(\alpha_y) & 0 & \cos(\alpha_y) \end{bmatrix}, \text{ where } \alpha_y \in (0,...,89)$$

$$R_z(\alpha_z) = \begin{bmatrix} \cos(\alpha_z) & -\sin(\alpha_z) & 0 \\ \sin(\alpha_z) & \cos(\alpha_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } \alpha_z \in (0,...,89) \qquad (6.10)$$

Rotation matrices are applied on geometrical representation of objects which consist a set of vertices (V_1,…, V_M). The general rotation matrix (Equation 6.11) is derived from multiplication of individual rotation matrices (Equation 6.10). The selected orientation of the object is then converted to voxel representation. The rotated object is obtained as follows:

$$\begin{bmatrix} x'_k \\ y'_k \\ z'_k \end{bmatrix} = \begin{bmatrix} \cos\alpha_y\cos\alpha_z & -\cos\alpha_x\sin\alpha_z + \sin\alpha_x\sin\alpha_y\cos\alpha_z & \sin\alpha_x\sin\alpha_z + \cos\alpha_x\sin\alpha_y\cos\alpha_z \\ \cos\alpha_y\sin\alpha_z & \cos\alpha_x\cos\alpha_z + \sin\alpha_x\sin\alpha_y\sin\alpha_z & -\sin\alpha_x\cos\alpha_z + \cos\alpha_x\sin\alpha_y\sin\alpha_z \\ -\sin\alpha_y & \sin\alpha_x\cos\alpha_y & \cos\alpha_x\cos\alpha_y \end{bmatrix} . \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} \qquad (6.11)$$

Where (x', y', z') denotes the rotated coordinates of the k^{th} vertex of the original object. The voxel size of each rotation is calculated as follows:

$$V_{\alpha_x,\alpha_y,\alpha_z} = [\max\Big|_{k=1}^{M}\{x_{k\alpha_x}\} - \min\Big|_{k=1}^{M}\{x_{k\alpha_x}\}]*[\max\Big|_{k=1}^{M}\{y_{k\alpha_y}\} - \min\Big|_{k=1}^{M}\{y_{k\alpha_y}\}]*[\max\Big|_{k=1}^{M}\{z_{k\alpha_z}\} - \min\Big|_{k=1}^{M}\{z_{k\alpha_z}\}] \qquad (6.12)$$

The minimum bounding box is derived from all produced voxel using:

$$V_{\min} = \min\Big|_{(\alpha_x,\alpha_y,\alpha_z)=0}^{89} V_{\alpha_x,\alpha_y,\alpha_z} \qquad (6.13)$$

Figure 6.3 shows the cross section of a 3D component that is reoriented in such a way that the minimum bounding box is generated. As a result, the voxel size is resized from 10x8x3 to a 7x7x3 unit. This orientation is used by the container loading algorithm which is described in the following sections. Since precast components have irregular shape and they must be stable within the container, the rotation angle domain is reduced to only 0, 45 and 90 degrees around each axis in the context of this study.

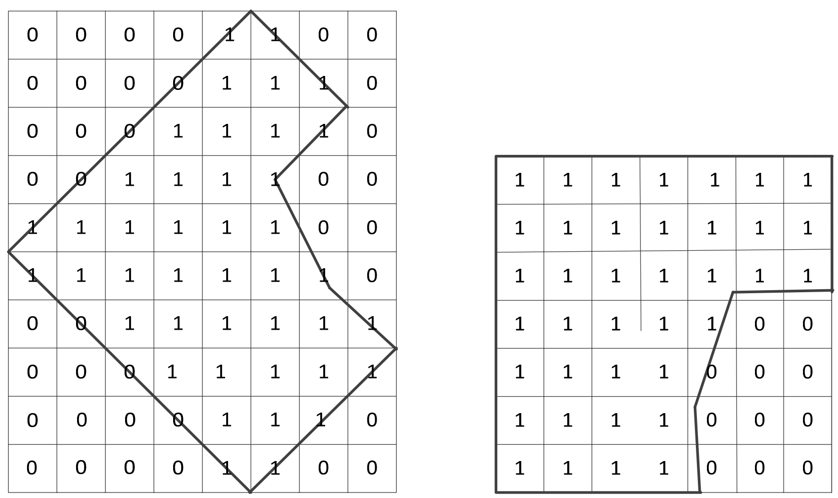


Figure 6.3: Object reorientation for finding minimum bounding box

6.4.2. Container Loading Algorithm (CLA)

Container Loading Algorithm is the core function of the proposed system. Container loading is a heuristic method that is developed to place the current prefabricated component into a truck without affecting any other located components. CLA minimizes the total transportation cost considering the loading strategy and application. Depending on the application, the problem can be modeled as either minimizing the total number of trucks which have fixed width, height and the length or minimizing the total length of truck or storage area which has fixed width and height and infinite length.

The flowchart of CLA is shown in Figure 6.4. The container loading is performed sequentially based on the order given by the sequence generator function. During each object placement, an incoming prefabricated component is placed as close as possible to the previously located components. In order to place components as close as possible to existing components in the truck, following equation is defined and termed as the objective function of the CLA. Equation 6.14 tries to place an incoming component along width, height, and length of the truck by minimizing the y, z, and x coordinate of the incoming component respectively.

$$minimize\text{: } f_{CLA} = x \times (W \times H) + z \times (W) + y \tag{6.14}$$

Where, "*x*" is the maximum coordinate of the component along X axis in truck's voxel. "*y*" is the maximum coordinate of the component along Y axis in the truck's voxel. "*z*" is the maximum coordinate of the component along Z axis of the truck's voxel. W and H are width and height of the truck. W and H are width and height of truck.

Although the component has been rotated around its local axes to obtain the minimum bounding voxel (explained in section 6.4.1.2); its global placement orientation into a truck's voxel may also produce different results. Therefore, different orientation of each component is checked by CLA to obtain the minimum unused space during container loading process. This rotation is called In-Place Rotation in Figure 6.4. Practically, components are not stable on odd angles (for example 26˚), thus only four orientations of components including 0˚, 90˚, 180˚ and 270˚ around x, y and z axes are checked for placement in truck. These angles are termed as $\theta_x, \theta_y,$ and θ_z, respectively. Therefore, Equation of 6.14 is calculated for all possible $\theta_x, \theta_y,$ and θ_z (64 angles) to find the minimum f_{CLA}.

Using Equation 6.14, when the truck is empty, the first component (no. 1 in Figure 6.5) is placed at lower left corner of the truck where x, y and z are minimized. The CLA tries to place following components along y until there is not enough space in this direction (see component no. 5 in Figure 6.5). Then a new row along z axis is formed to place next components (no. 6 in Figure 6.5). When there is not enough space along y and z axis, then a new row is formed along x axis for new component placement (see component no. 8 in Figure 6.5). The sliding procedure is one unit of the voxel in one direction at the time. The sliding procedure is terminated if one of the following conditions occurs; new move produces overlap, or the resulting component cannot be fit within the current container.

The stability criteria of the component is checked when the placement of the component is determined in the truck. The sliding procedure is performed for all possible In-Place rotation angles but only the one with the minimum CLA objective function is selected as the final solution.

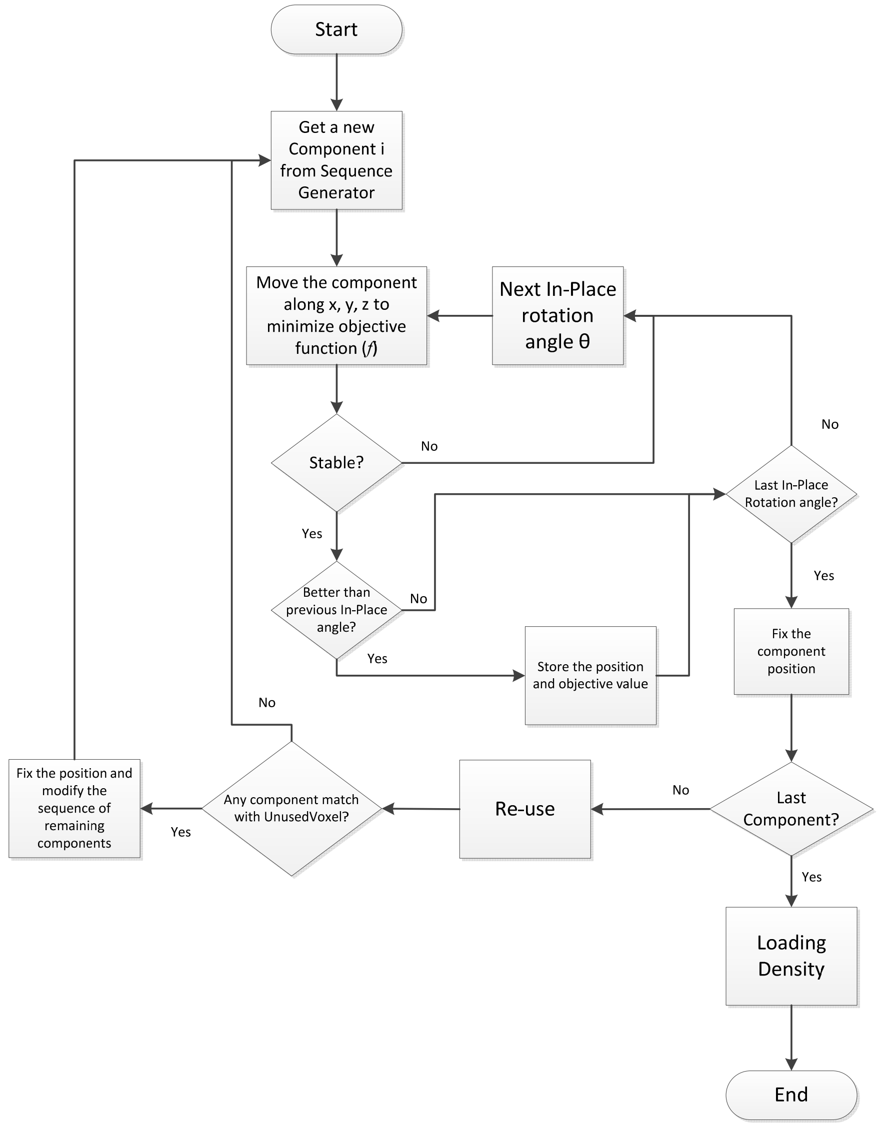


Figure 6.4: Container Loading Algorithm flowchart

However, many free spaces may be produced as a result of this process. These unused spaces might be reconsidered as an allocation space if it is large enough to accommodate the remaining component(s). This will result in a higher loading density value of the placement procedure at that stage. The generated unused spaces must be detected and compared with the incoming objects to find a match. If there is a possible match, the selected space is used for the object placement; otherwise, the CLA is continued. An algorithm to use the residual space is proposed in the next section. Before implementing the reuse algorithm or next round of CLA, the total weight of the truck must be checked to avoid over-loading. It should be mentioned that such a high packing density may not be practical for concrete precast components. However, the prefabricated components are not restrained to any material and shape. Thus the algorithm allows timber, steel or light-weight gypsum component to be produced off-site and transported for assembly.

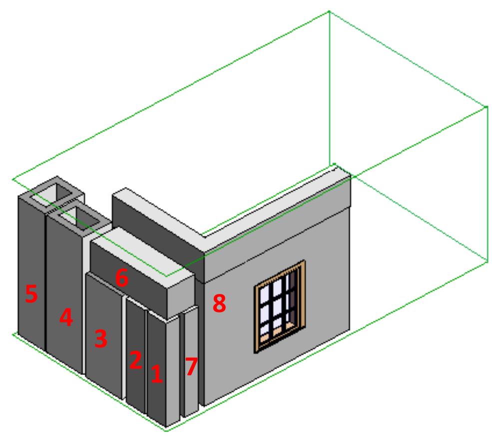


Figure 6.5: CLA placement order and orientation within a truck

6.4.2.1. Reuse Algorithm

In the proposed CLA, unused spaces are produced by previous components which are placed in the truck (Figure 6.6). These spaces may be reused to achieve a higher loading density of the truck loading. A "Reuse" algorithm is required to consider sequence of object placement, sequence of unloading, and accessibility to free spaces for transportation of building elements.

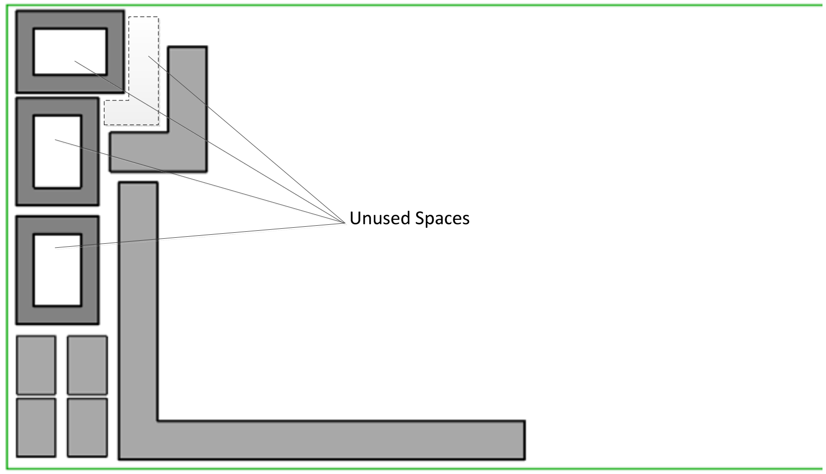


Figure 6.6: Unused space among located components within a truck

In the context of this study, an object can be placed between located objects when there is an opening toward z axis to the top. Thus, the crane would be able to place an object to the container.

The truck voxel which is used for reuse algorithm includes all the components that located in the truck. The space used by located objects is indicated by “1” in the voxel representation. The remaining unused space which is the complement of the located objects is represented by “0”. An algorithm is developed to fill unused space among located components in the container as follows:

1. Find the highest occupied level in the container (z_{max})

2. Scan the plane of Z_{max} to find zero elements in the container voxel and save it as two-dimensional matrix (Figure 6.7)

3. Check the “0” elements to be confined within “1” elements

4. Go to next row of the container voxel (z=z-1)

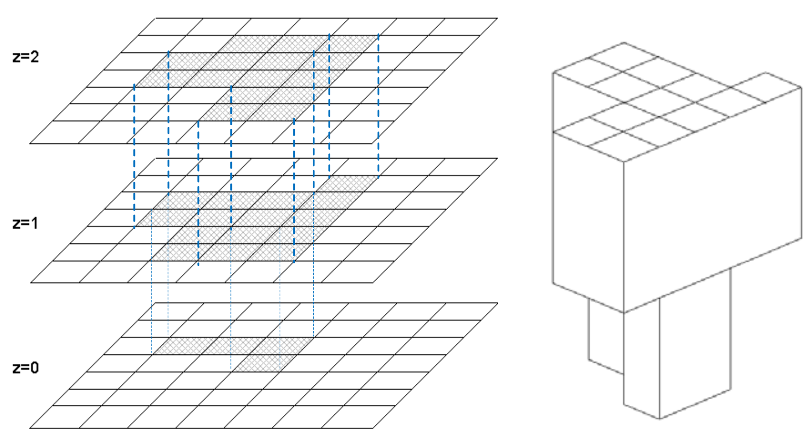
5. Scan the plane of z to find zero elements in the container voxel and save it as two-dimensional matrix

6. Repeat step 4 and 5 till reach the bottom of the container (Figure 6.7a)

7. Match 2D matrices to obtain the common area between each subsequent levels and create a temporary UnusedVoxel (Figure 6.7b).

8. Perform the Equation 6.14 for UnusedVoxel (UnusedVoxel is assumed as a truck and the sliding of objects is performed to place the biggest object among remaining objects into the UnusedVoxl)

9. Modify the sequence of remaining objects accordingly.



(a) (b)

Figure 6.7: Scanning truck voxel to find the temporary UnusedVoxel

6.4.3. Sequence Generator

The third module in the proposed system for container loading is the Sequence Generator. Finding an optimal or near optimal solution for container loading in the proposed model highly depends on the sequence of component placement. Different sequences of the same set of components may result in different packing densities. This is a permutation problem which can be modeled as the well-known traveling salesman problem. The sequence of the input components given to the algorithm can be mapped to all the cities traveled by the traveling salesman. The total space used in the container is similar to the total cost of the salesman's travel. However, this traveling salesman problem cannot be solved by conventional optimization methods because the objective function is non-linear. Genetic algorithm which is commonly used for these types of problems is selected as a solution technique for sequence generator to solve this problem. The sequence generator maps each possible sequence into a string (chromosome). Each element of the string is an index for components. Position of the element in the string corresponds to the index of the corresponding component in the input sequence. The initial population is randomly created. The fitness function of each sequence string is the transportation cost according to the loading strategy of the corresponding sequence obtained by the container loading. Subsequent generations or offsprings are repeatedly generated through genetic operators until a satisfactory result is achieved.

6.4.3.1. Genetic operators

In a permutation problem, every element of the problem must exist in each possible solution, but may be in different positions. In the standard crossover operators a random crosspoint is selected and the substring of the parents is exchanged to form the offsprings. Since in standard crossover every element of the parent may not exist in the generated offspring and some of the elements may exist more than once, this crossover is not suitable for the present permutation problem. This difficulty also exists for standard mutation operators. In the standard mutation operator, some randomly selected genes are flipped (ones changed to zeros, zeros changed to ones).

Literature review shows that several crossover and mutation operators have been developed for permutation problems. Partially Mapped Crossover (PMC) developed by Goldberg and Lingle (1990), Order-crossover by Davis (1985), Order based crossover by Syswerda (1991), Position based crossover by Syswerda (1991) and Heuristic crossover by Grefenstette (1985) are commonly used crossover for permutation problems.

According to Larrañaga et al. (1999) order based crossover (OX1) proposed by Davis (1985) shows better performance, thus OX1 is selected as the crossover operator for the proposed sequence generator module.

The OX1 employs characteristics of path representation in which the order of objects is important rather than their position. Using OX1 crossover operator, an offspring of one parent is generated by choosing a substring; while the relative order of the second parent is preserved. For example, the following strings demonstrate two parents of the current generation:

Parent 1: (7 2 5 4 3 6 1 8) and

Parent 2: (1 2 6 3 7 8 4 5);

Two random cut points are required to select a sub-string from the parents. In the following example the first cut point is placed between the second and the third object and the second cut point is placed the fifth and the sixth object.

Parent 1: (7 2 | 5 4 3 | 6 1 8) and

Parent 2: (1 2 | 6 3 7 | 8 4 5):

Therefore, as can be seen in the following, to create the offsprings of the next generation, the sub-strings of parents are copied to the offspring. i.e. an offspring inherits the sub-string from the first parent.

Offspring 1: (** | 5 4 3 | ***) and

Offspring 2: (** | 6 3 7 | ***)

Starting with the first entry of parent 2, the objects are inserted into offspring 1 beginning after the second cut-point. When the end of offspring1 is reached, the insertion continues at start at start of offspring 1. If the object already exists in offspring 1 it is not inserted. This procedure is demonstrated for the given strings in the Figure 6.8 where Parent 1, and Parent 2 are transformed to:

Offspring 1: (7 8 | 5 4 3 | 1 6 2) and

Offspring 2: (1 8 | 6 2 7 | 5 4 3);

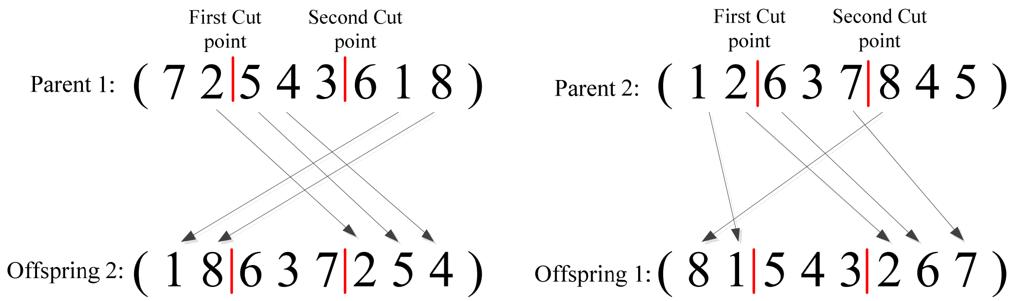


Figure 6.8: OX1 crossover operator

In order to facilitate searching into new search space, another mutation operator is required. Several mutation operators have also been developed for permutation problems. Among the developed mutation operators, Simple inversion mutation by Holland (1975), Insertion mutation by Fogel (1988), Exchange mutation by Banzhaf (1990), Inversion mutation by Fogel (1990) and Scramble mutation by Syswerda (1991), and Displacement mutation by Michalewizc (1996) are the most well-known mutation operators

According to the literature review provided by Larrañaga et al. (1999), the Displacement Mutation (DM) is selected in this study because it shows the best performance among the mutation operators. The DM operator developed by (Michalewicz, 1996), initially selects two random cut points so that a sub-string is produced. This sub-string is removed from the string and inserted in a random place accordingly. The DM mutation is explained through the following given string.

String 1: (1 2 3 4 5 6 7 8)

The sub-string (345) is produced by two random cut-points (one before third object and the other one between the fifth and the sixth objects i.e. (12|345|678)). The remaining objects after removing the selected sub-string are as follows:

String 2: (1 2 6 7 8)

Finally a random insertion point is selected for the omitted sub-string (For example, the position between 7 & 8 in the string 2). Thus the following is the string that is produced after performing mutation operation:

(1 2 6 7 3 4 5 8)

6.4.3.2. Selection Strategy

The Binary Tournament (BT) selection strategy is adopted in this study due to the advantages reported by Blickle and Thiele (1996) including: efficiency and ease of implementation, efficient time complexity, especially if implemented in parallel, low susceptibility to takeover by dominant individuals, and no requirement for fitness scaling or sorting.

The mechanism of BT selection is shown in Figure 6.9. In BT selection, 2 strings are selected randomly from the current population, and are compared against the other. The string with a higher value of fitness function is transferred to the population of next generation. The BT preserves the diversity of the solution set because of equal chance of selection. However, maintaining the diversity may degrade the convergence speed.

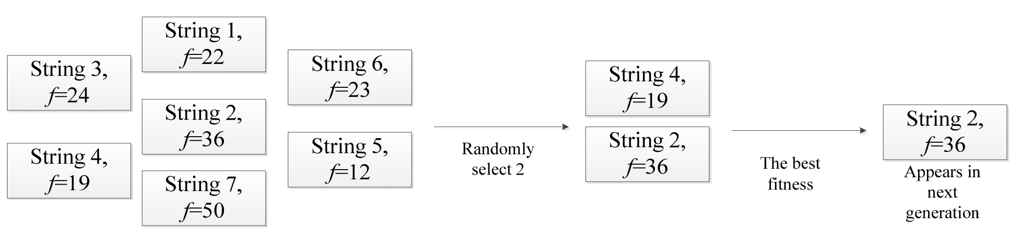


Figure 6.9: Mechanism of Binary Tournament selection strategy

In order to keep the best solution in previous generations, 2% of the strings with the highest fitness value are duplicated to next generation. Remaining strings (98%) are selected using BT.

6.5. Implementation of the model

For better integration with the previous systems that were developed in this study, the proposed model for container loading problem is also developed using C#.NET platform. The implementation of the three main modules of the proposed system together with a visualization module that is developed for understanding the results is discussed in the following paragraphs.

6.5.1. Preprocessor Module

The preprocessor module receives the list of component for each prefabrication configuration from the configuration generator module (See Chapter 3). Since B-rep schema is used for component representation in this study, the given components are represented by set of vertices at their lower level of their representation method. The set of vertices are utilized in Preprocessor module for individual object orientation. This module also uses general rotation matrix to obtain the minimum bounding box. The voxel representation of the prefabricated components is achieved using the described method earlier in this chapter. Having the geometric representation, the Preprocessor module calculates the center of gravity to obtain the stable orientation of prefabricated components. Unstable orientations are omitted to reduce computation time. Then after, the preprocessor module assigns the required margins for accessibility, lifting points and separators. Finally, it generates both geometric and voxel representations of 3D components according to the object's new local orientation.

6.5.2. Sequence generator module

Genetic algorithm is exploited to enhance the performance of the system by generating new sequences for performing better arrangement by CLA. The number of sequences used in each problem is varied based on the size of the problem that is provided in an input configuration list. The first population is randomly generated.

As stated earlier in the model description, the selected genetic operators used in this research are reproduction, OX1 and DM. Binary Tournament is applied to the old population to create their offsprings. The best 2% within the current population are preserved for the next population, so the best solution found is always kept within the population. Crossover and mutation rate used by the OX1 and DM operator are determined as input parameters to the system. The rates of 0.6-0.8, and 0.2-0.4 are used for the OX1 and DM respectively to evaluate the performance of the proposed model.

6.5.3. Container Loading Algorithm

This module is the core module of the system. It performs the placement procedure on list of components provided by preprocessor based on a sequence given by the sequence generator. Dimension of trucks, maximum weight are given as constraints to this module.

Initially, the CLA locates the component into the far end of the components located in the truck. Then it tries to relocate the object toward the left and lower bottom position of the truck to reduce gaps between the object itself and the located objects within the truck so the total required number of containers can be reduced. Steps and constraints used to relocate the component are stated in the previous section. The time required for running CLA on each component is relatively high. However, since the free space in the truck is reduced as the number of located components increases, the total computation time does not increase. Once the current truck is filled, it is stored in database and a new truck voxel is created for remaining objects. When the process is finished, the global coordinates of the component is updated and referenced to its new location.

6.5.4. Visualization Module

Assessment of the obtained result is a challenging task due to proposed representation schema. Three-dimensional matrices cannot be presented on paper or screen. Since the voxel representation is used in this study to represent irregular-shaped objects, a parser module is developed to convert filled-truck's voxel to a graphical environment. 3D rendering is done with Hoops 3D framework developed by TECHSOFT3D Co.

6.6. Experimental results

To study the performance of the proposed model, a benchmark analysis and two case studies with various scenarios are considered. The developed prototype is executed on Dell Precision T5500 with Intel(R) Xeon (R) CPU X5650 @ 2.67 GHz Processor with 48 GB of RAM running Windows 7 Professional 64-bit operating system.

The case studies in this chapter are aimed to: 1) verify the performance of the proposed model in terms of loading density for 3D irregular-shaped objects, and 2) provide a comparison on transportation cost of conventional and new prefabrication configuration approaches in terms of number of trucks that is required.

6.6.1. Benchmark Analysis

Since there is no benchmark for real irregular shaped (strongly heterogeneous non-convex) container loading problem, the validation is performed only for 3D cuboid objects. In order to evaluate the obtained results from the proposed model, 7 benchmark problems are selected from the benchmarks generated by Bischoff and Ratcliff (1995). This set of problems is the most commonly used in the literature. The benchmarks are classified from easy (Bench1) to difficult (Bench7). These categories are different in cuboid object type, size and numbers. There are no constraints involved. The presented results are averaged.

The results are compared against the following algorithms reported in the literature: 1) CBGAT by (1997) 2) The GRASP approach by Parreno et al.(2008), 3) the tabu search algorithm CBUSE created by Bortfeldt and Gehring (1998), and 4) the algorithm created by Bischoff and Ratcliff (1995).

Table 6.1 presents the results of benchmark analysis. The first three columns presents the average results obtained from previous algorithms which are reported in Bortfeldt and Gehring (2002). The GRASP and the tabu search algorithm CBUSE achieve the best results to date, with the GRASP algorithm achieving the highest average volume utilization across all benchmark problems. The proposed model in this study does not outperform CBGAT, CBUSE and GRASP. However, obtained results by VoxelCLA show a higher packing density in comparison to Bischoff and Ratliff method.

Table 6.1: Loading density of benchmark analysis from Bischoff and Ratcliff (2002)

Problem		Bischoff & Ratcliff	CBGAT	CBUSE	GRASP	VoxelCLA
Name	Number of Type					
Bench1	3	83.37	85.80	92.41	93.85	85.89
Bench2	5	83.57	87.26	92.33	94.22	87.25
Bench3	8	83.59	88.10	91.57	94.25	87.10
Bench4	10	84.16	88.04	91.26	94.09	84.78
Bench5	12	83.89	87.86	90.40	93.87	81.91
Bench6	15	82.92	87.85	89.57	93.52	84.46
Bench7	20	82.14	87.68	88.18	92.94	83.97
Average		83.38	87.51	90.87	93.82	85.05

6.6.2. Case Studies

In order to evaluate the transportation cost of the traditional prefabrication method and a higher level of prefabrication, the Example 2 discussed in Chapter 5, is studied. The conventional approach in precast production carries regular (mainly rectangular) shaped objects. However, a higher level of prefabrication deals with both cuboid and irregular-shaped components.

Experiments are performed for different crossover rates (0.6 - 0.8) and different mutation rates (0.2 - 0.4). Population sizes are set up to five times of the number of required objects.

6.6.2.1. Cuboid objects

The first case study is designed to evaluate the proposed method for 3D cuboid objects which represents the conventional approach in prefabrication of building elements. The problem is the minimum space required for 391 cuboid components created from 8 different component types.

Stockyard Scenario (infinite length)

In the first scenario, these components are located into a container with a finite width and height of 5 and 4 meters respectively but infinite in length (e.g. Stockyard) so that weight constraint is not considered. A fixed population size of 1200 with different crossover and mutation rates are utilized in this scenario. One hundred generations are produced for each test run. The results are shown in Table 6.2.

Table 6.2: Loading Capacity of Stockyard for 3D cuboid components due to various GA parameters

Crossover rates	Mutation rate		
	0.19	0.25	0.38
0.6	72.36	79.91	79.23
0.7	74.69	80.36	78.67
0.8	76.44	78.72	77.57

The maximum loading density of 80.36% is achieved with 70% crossover and 25% mutation rates in this scenario. Although the problem has lower number of component types and numbers and also the components are huge and bulky, the result indicates that the quality of packing in stockyard when the weight of objects is not considered is in the range of previous CLP approaches. Better packing density could be achieved by removing the constraints of this problem such as margins for lifting and accessibility.

Transportation Scenario (finite length)

The second scenario for this case determines the minimum number of trucks that is required for transportation of all components. The weight of component and loading/unloading access are considered in this scenario so that the loading strategy is constant distribution. The transportation cost of prefabricated components per truck is $100. The result obtained with the same set of GA parameters indicates that 43 trucks must be employed. The average weight of trucks is 23.5 tons which shows that almost 94% of truck loading capacity is utilized. The weight distribution over the trucks is represented in Figure 6.9. The total cost of transportation of the component using the traditional prefabrication method is $4300.

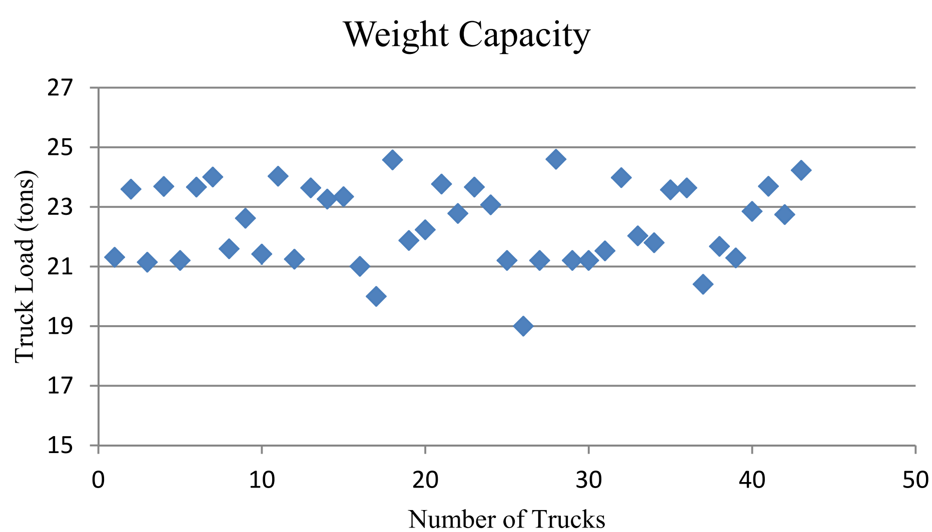


Figure 6.10: The loading distribution of trucks for cuboid components

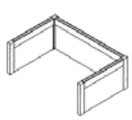
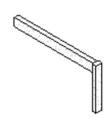
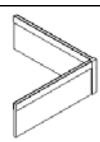
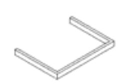
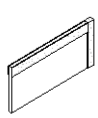
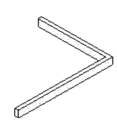
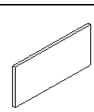
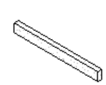
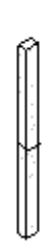
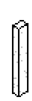
Transportation Scenario (finite length & light components)

The third scenario tries to find the minimum number of trucks when the components are produced from light material so that the weight of components does not govern the constraints. Results that obtained from the same set of GA operators show that the minimum number of truck which is required to transfer all 391 components to the construction site is 7 trucks. The average occupied space in trucks is 74.33% which is 9% lower than the infinite length scenario. Since the components are not designed to fit in a certain number of containers properly and also containers have limited width, height, and length, the loading density is lower than the first scenario.

6.6.2.2. Irregular-shaped objects

The second case study is designed to minimize the number of trucks to transport the required components for construction of the residential building that described in Chapter 5 using a higher level of prefabrication. The selected configuration of prefabricated components comprises 10 types of components. The required number of each component, dimensions and their weight are shown in Table 6.3.

Table 6.3: Component Types and Characteristics for Transportation

No.	Shape	Qty.	Volume (m^3)	Weight (tons)	No.	Shape	Qty.	Volume (m^3)	Weight (tons)
1		12	17.55	24	6		12	2	4.4
2		10	10.45	15.1	7		6	3.75	8.2
3		13	5.6	8.7	8		10	2.5	5.5
4		20	3.6	3.6	9		40	1.25	2.7
5		15	1.5	3.3	10		30	0.75	1.6

Stockyard Scenario (infinite length)

In this case study, the first scenario is aimed to evaluate the performance of the developed algorithm to obtain the maximum loading density for 168 strongly heterogonous 3D components. To this end, it is assumed that the container has limited width and height but unlimited length (e.g. stockyard). The weight constraint is not considered in this first instance. Different GA parameter sets including crossover and mutation are used to run this scenario. Five hundred generations are produced for each test run. The obtained results are summarized in Table 6.4. The best packing density is 76.43% using 80% crossover and 25% mutation rate. The loading density for 3D irregular shapes is 10% lower than the cuboid objects.

Table 6.4: Loading Capacity of Stockyard for 3D irregular-shaped components due to various GA parameters

Crossover rates	Mutation rate		
	0.2	0.3	0.4
0.6	63.26	75.36	60.51
0.7	65.98	68.91	64.25
0.8	65.90	77.43	67.10

Transportation Scenario (finite length)

The second scenario describes the practical situation in which 168 components are moved by the minimum trucks using constant packing strategy. The truck dimension is 5 by 4 by 12 meters for width, height and length respectively. The maximum loading capacity of trucks is 25 tons. The transportation cost of prefabricated components per truck is $100. Different GA parameter sets including crossover and mutation are used to run the second scenario. The number of required trucks is 50 and the transportation cost of prefabricated components for this project is $5000. In this scenario weight of components is the governing constraint rather than the space. For example in this scenario, each component of type 1 must be transported with 1 truck. The average weight of trucks is 20.33 tons which shows that almost 81.32% of truck loading capacity is utilized. The weight distribution over the trucks is represented in Figure 6.10.

This study shows that although the transportation of the prefabricated components using a higher degree of prefabrication requires 50 trucks (traditional method requires 43 trucks). Transportation of irregular components is $700 more expensive than the traditional approach for each construction cycle. The transportation cost of the new method of prefabrication is still in an acceptable range due to small difference between obtained results.

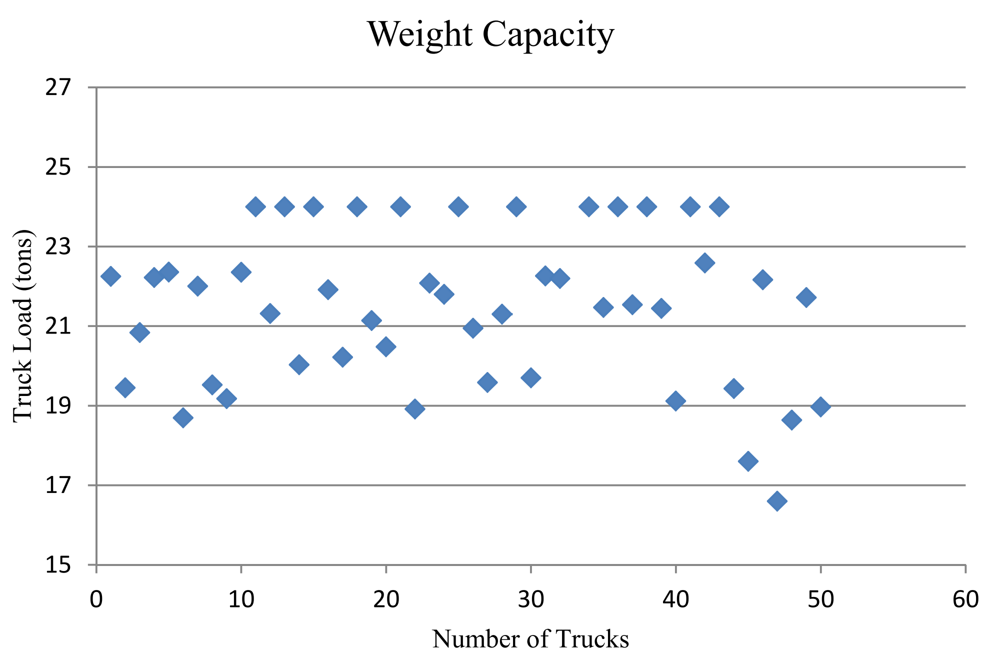


Figure 6.11: The loading distribution of trucks for 3D irregular-shape components

Transportation Scenario (finite length & light components)

The third scenario of this example is aimed to find the minimum number of trucks when the components are produced from light material so that the weight of components does not govern the constraints. The same set of GA operators has been used to run this scenario. Results show that the minimum number of truck which is required to transfer all 168 components to the construction site is 11 trucks. These components require 7 trucks for transportation if they are produced individually (cuboid). The average occupied space in trucks is 63.76% which is 10.57% lower than light material cuboid scenario. Because in this scenario number of trucks is determined by, dimension of component type 1, its stability and loading-unloading direction.

The last analysis in this chapter is designed to highlight the importance of the resource optimization for transportation of the produced components to the construction site. In this analysis 100 random sequences of components are utilized for container loading for both conventional approach and selected prefabrication configuration (Case study 1 and 2). The real weight of components and the container capacity are considered in this scenario. The average number of trucks that must be adopted for conventional approach is 58 while the proposed model optimized the number of truck to 43. The number of required trucks for the selected prefabrication configuration is 65 while the proposed model obtained 50 trucks. Apart from the prefabrication configuration, these results indicate that the proposed model is able to reduce the transportation cost by more than 30%.

6.7. Concluding Remarks

In this chapter, a new emerging approach for a particular form of Container Loading Problem was illustrated. The problem was modified for the main objective of this research which is minimizing the resource required to employ a higher degree of prefabrication. Thus the proposed model attempts to obtain the minimum number of trucks required for transportation of 3D irregular-shaped prefabricated components.

Many solution approaches for container loading problem, which is one form of the general bin packing problems were reviewed and commented on, but as yet, there is no single powerful algorithms that can solve this problem effectively. Heuristic and genetic algorithms are found to be useful tools. A new approach that combines the use of both techniques is proposed as a solution for both rectangular-shaped and irregular-shaped container loading problems. In order to overcome the challenges of presenting 3D irregular–shaped components for the proposed algorithm, a novel Volumetric Pixel (Voxel) presentation is developed.

The proposed system consists of three main components: a Preprocessor module, a Container Loading Algorithm (CLA) and a Sequence Generator. The input module is an interface between the CoCoPS framework and the proposed system in this chapter. The CLA performs the detail placement layout based on the sequence of objects given by the sequence generator. It sequentially places each object in such a way that the resulting unused space is minimized. Since the prefabricated components are usually heavy and bulky, a constant weight distribution loading strategy is defined to have easy loading/unloading access.

The model was verified by the benchmarks created by Bischoff and Ratcliffe (1995). The obtained results showed that the proposed algorithm does not outperform CBGAT, CBUSE and GRASP but our model gives a higher packing density in comparison to the model developed by Bischoff and Ratcliffe (1995). Since there is no benchmark for strongly heterogonous 3D objects in the literature, the verification is done only for cuboid objects.

Two case studies with several scenarios and varying parameter sets are designed on the same building to evaluate the transportation cost of prefabricated components. The first case is designed to obtain the optimal arrangement of components on trucks for traditional prefabrication approach. The second case study is designed to find the optimal arrangement of components for a higher degree of prefabrication.

The main finding of this model indicates that the developed model is able to reduce the transportation cost by 30% for both irregular and cuboid components compare with random container loading. Further to this, the result shows that although the selected configuration is more expensive in transportation cost, it can be still feasible due to small difference between obtained results which is \$5000 for the selected configuration and \$4300 for the conventional approach.

In stockyard problem scenario where width and height are finite and length of the stockyard is infinite the loading capacity of 80.36% and 76.43% was obtained for Cuboid and Irregular-shaped components. Although the building components for traditional prefabrication are cuboid, average loading density is lower than the results obtained from benchmark analysis. This is because of constant packing strategy, margins, accessibility, number and size of components in this study.

For transportation scenario in which length, width and height are finite, the number of trucks required to transport prefabricated components considering the weight of components is 43 and 50 for traditional and higher degree of prefabrication respectively. This analysis also indicated that the algorithm could fill up trucks with the average of 23.5 and 20.33 tons out of 25 for cuboid and irregular-shape objects respectively.

Although the obtained results were satisfactory; the proposed approach can be improved by considering more real constraints (such as load bearing capacity of underneath objects, loading and unloading access) in the model.

CHAPTER 7: CASE STUDY

This chapter presents a case study for illustrating the application of the developed concepts, algorithms, optimization models, and analysis tools. In this study, an automated system is developed to integrate the designer, producer and constructor to adopt a higher degree of prefabrication in a project. In other words, this study is meant to move beyond traditional individual element prefabrication approach towards grouping building elements to employ a higher degree of prefabrication. Based on these discussions, this chapter will demonstrate: 1) the functionality of the proposed CoCoPS framework through a real project, 2) the challenges that may be encountered during the implementation on large scale projects, and 3) show the economic advantages of adopting a higher degree of prefabrication in a real project.

The project client, or fabrication contractor may use CoCoPS to reduce project time, and cost. This chapter demonstrates how the proposed framework is able to retrieve required data from 3D model using IFC, deduce topological relationships among building elements, create GDM, generate feasible configurations and find the optimal configuration of building elements for prefabrication. This case study includes:

1. Importing building elements' data including ID, type, geometric representation, dimension, and material from IFC,
2. Unifying geometric representation of building elements,
3. Finding topological relationships among building elements using the proposed deduction algorithm,
4. Mapping building elements and relationships to the proposed GDM,
5. Performing graph algorithms to generate a set of feasible configurations,

6. Finding the optimal feasible configuration in terms of mould usage, transportation, and installation cost.

The objective of this case study is to demonstrate that the proposed framework can reduce the project resource utilization cost in comparison with traditional method in prefabrication.

7.1. Case Study Description

The selected residential building is one of the biggest residential complexes designed for the National Housing Development programme in Iran, which presents the Iranian innovations and affirms to foster construction methods. It is called “Diamond Town” and it is planned to finish by 2015.

This town comprises 91 blocks of residential towers, one town centre, one sport complex, one educational complex, and 3 parking towers. The schematic site layout is presented in Figure 7.1.

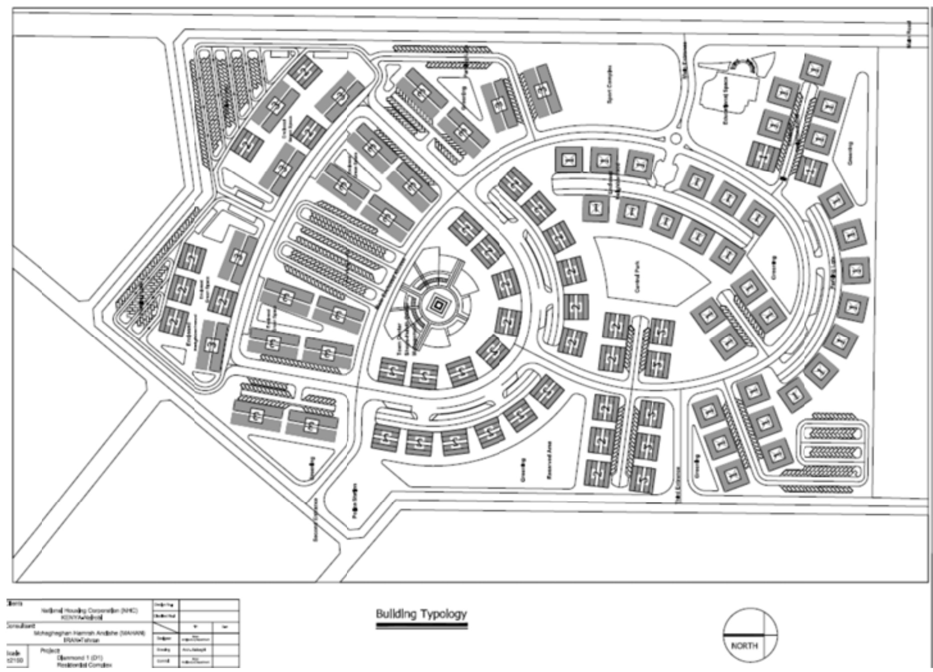


Figure 7.1: Site Layout of “Diamond” town

The total area of Diamond Town is 231,644 m^2. The total area occupied by residential buildings is 52,000 m^2.

Residential buildings in this complex consist of 3 types of blocks naming as “1”, “2”, and “3”. Each type has different layout plans and heights. There are six typical units in this project which are called A, B, C, D, E and F as demonstrated in Figure 7.2.

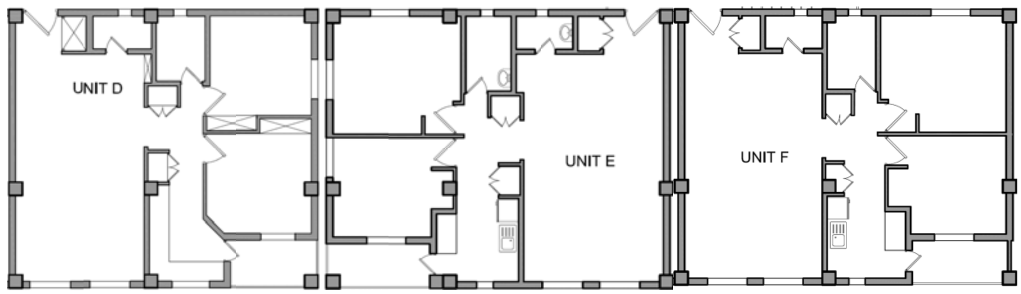


Figure 7.2: Floor plan of 6 typical units

The first block type "1" is a 10-storey building with two types of units (A and B) each of which are 3 bedroom flats. The block has four units per level. There are 35 of this type in the town. The total area occupied by Type "1" is 17,690 m^2. Plan of this type is shown in Figure 7.3.

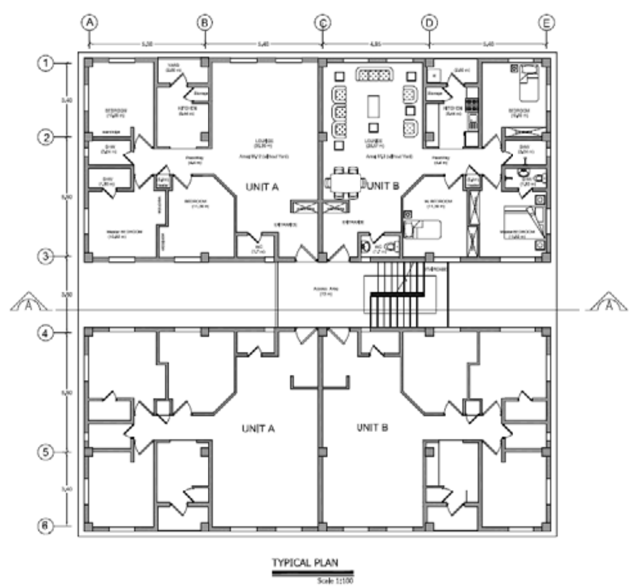


Figure 7.3: Floor plan of Block type "1"

The second type of residential block "2" comprises units C and D. There are 2Cs and 2Ds units per level. Units C and D are 2-bedroom flats. The total area occupied by this type is 14,393 m^2. There are 37 of this block type in the project. The plan of this block is shown in Figure 7.4.

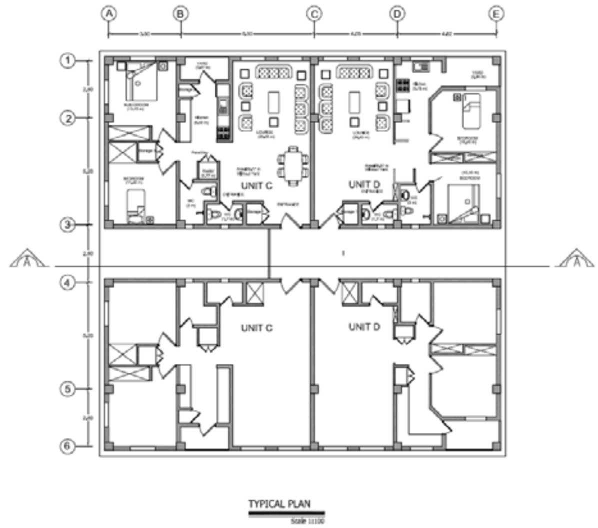


Figure 7.4: Floor plan of Block type “2”

The last block type in this case study, which has the largest floor area, comprising four flats of type E and four of type F in each level. These unit types are 2-bedroom apartments. The total area of this block “3” is 15,770 m^2. Table 7.1 summarizes the block types, unit types and number of flats in the town. The floor plan of block type “3” is shown in Figure 7.5.

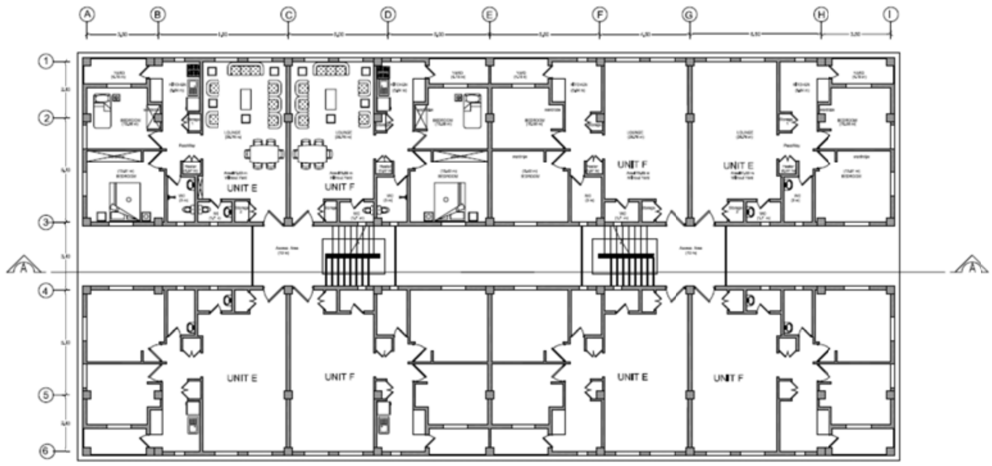


Figure 7.5: Floor plan of Block type "3"

Table 7.1: Project information summary

Block Type	Number of Storeys	Number of blocks in project	Unit types in each block	Number of units in each block	Area of Block	Total are of this block in town	Total Area of Flat Contraction
1	10	35	A,B	4	490	17690	176900
2	10	37	C,D	4	389	14393	143930
3	10	19	E,F	8	830	15770	157700
Summary	-	**91**	-	-	-	-	**473130**

7.2. CoCoPS Framework Implementation for Diamond Town

The detail implementation of the CoCoPS framework was illustrated in Chapter 3. The case study presented in Chapter 5 has repetitive floor plans. However, the Diamond Town project has three different block types with different floor plans. Therefore in order to achieve a higher degree of prefabrication in this project, configuration generator, comparator and constructability function in CoCoPS must analyze the whole project at the same time. Other modules and functions such as Preprocessing and topological relationship deduction are performed on each block type to reduce calculation time and complexity. The optimization on production resources is performed on all block types at the same time; while transportation optimization is performed for on each block type because of various destinations of trucks.

7.2.1. Creating 3D model and IFC file with Autodesk Revit 2012

The project model is created using BIM software. The data transfer from 3D model to IFC files could be achieved automatically by IFC-compatible CAD applications. In this project, Autodesk Revit Structure 2012 with its IFC2x Utility is used as the tool to automatically generate an IFC file from the 3D model. A screen capture of the 3D model of the Diamond Town and residential block type “1” are shown in Figure 7.6.

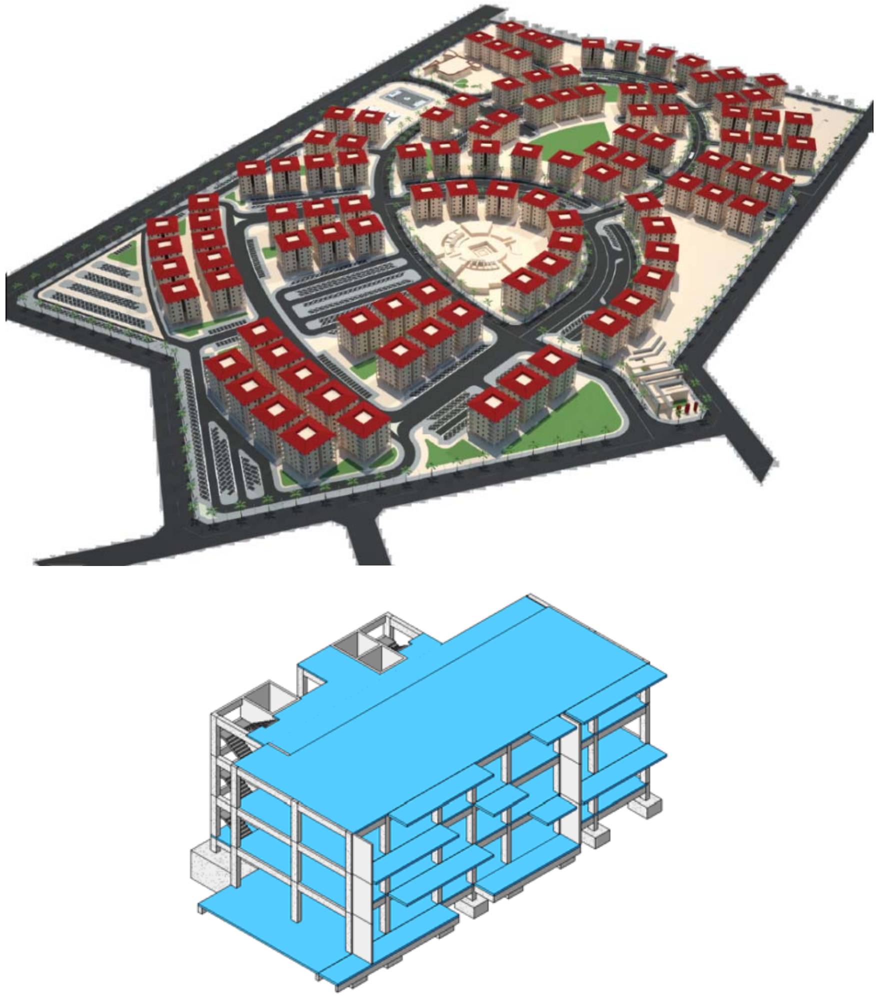


Figure 7.6: a) 3D view of the project layout, b) 3D section of the residential building

7.2.2. Extracting building elements and required data

The second module in the proposed CoCoPS framework is Preprocessor in which building elements and their properties are extracted from IFC file and are stored into the designated internal data structure. The preprocessor also allows the users to manipulate the properties of imported elements. In this case study column, beam, wall, and slab are the elements that are retrieved from IFC.

Beam, column, wall and slab are recognized through the IfcBeam, IfcColumn, IfcWallStandarsCase, and IfcSlab in the IFC file respectively. The data acquired from IFC for each building element includes: ID, Type, Name, Material, Dimension, geometric representation, topological representation (if any) and their location in the 3D model. Having this data, the preprocessor module calculates area, volume, weight, and global placement of building elements. Table 7.2 and Table 7.3 show the statistics of building elements that has been obtained from IFC file of Diamond town model.

Table 7.2: Building elements per block type

Block Type	Number of Column per level	Number of Beams per level	Number of Structural Walls per level	Number of Partitioning Walls per level	Number of Slabs per level
1	30	44	28	84	24
2	30	44	28	64	20
3	54	84	60	168	44

Table 7.3: Total building elements extracted for Diamond town project

Block Type	Number of Columns per Block	Total Number of Columns	Number of Beams per Block	Total Number of Beams	Number of Structural Walls per Block	Total Number of Structural Walls	Number of Partitioning Walls per Block	Total Number of Partitioning Walls	Number of Slabs per Block	Total Number of Slabs
1	300	9900	440	14520	280	9240	840	27720	240	7920
2	300	11100	440	16280	280	10360	640	23680	200	7400
3	540	10260	840	15960	600	11400	1680	31920	440	8360

7.2.3. Generating Feasible Configurations

7.2.3.1. Developing GDM

In order to create a GDM for this case study, it is required to reduce the size of graph by avoiding repeated floor plans. Since the real constraints in production, transportation and installation do not allow having components which are bigger than 10m in length, 5m in width, and 4 meter in height, the proposed GDM is built for two storeys of each block type. Table 7.4 shows the number of building elements which forms the set of nodes in GDM. The generated graphs comprise 420, 372, and 820 nodes for blocks "1", "2", and "3" respectively.

Table 7.4: Statistics of nodes in GDM

Block Type	Column Nodes	Beams Nodes	Structural Wall Nodes	Partitioning Wall Nodes	Slab Nodes	Total Number of Nodes
1	60	88	56	168	48	420
2	60	88	56	128	40	372
3	108	168	120	336	88	820

The Constructability Analyzer module deduces the topological relationships among building elements using the proposed algorithm. By this, the GDM has 1459, 3845, 8456 edges for blocks "1", "2", and "3", respectively. The solution time and accuracy are examined to validate the proposed algorithm for the proposed IFC-based topological deduction algorithm. In order to evaluate the calculation time of the proposed algorithm for deduction of topological relationship among building elements, the result is compared against ISO19107 spatial schema that was described earlier in Chapter 4.

Figure 7.7 presents the total processing time for all four relation (Connectivity, Containment, Intersection, and Separation) queries using the two data models with the x-axis denoting the number of building elements in the GDM. The query times between the two data models start to show a significant difference even for the smallest graph size which belongs to type "2" with 380 nodes. It takes 830 seconds to run the query and deduce all four types of topological relationships among building elements whereas it takes 2569 seconds using conventional ISO19107 spatial schema. For type "1" with 428 building elements, it takes 4264 seconds using conventional ISO19107 spatial schema whereas it takes only 1215 seconds using GDM. The computation time for running the query for the largest block type, "3", (with 820 building elements), is 3200 seconds while the same query using ISO19107 takes 15903 seconds (more than 45 days). The reason is that in conventional ISO19107 schema, all building elements and their geometric information are stored in a relational database. This method requires performing binary check on building elements for all four types of topological relationships using the geometric data. It can be concluded that the computation cost for conventional ISO19107 approach increases exponentially to find topological relationships among building elements. On the other hand, the computation using GDM does not show any exponential increase with elements. Thus GDM is much more efficient than the ISO 19107 Spatial Schema for topological relationship queries. Moreover, this analysis shows the performance of the GDM in handling large-scale topological representation in micro-spatial environments.

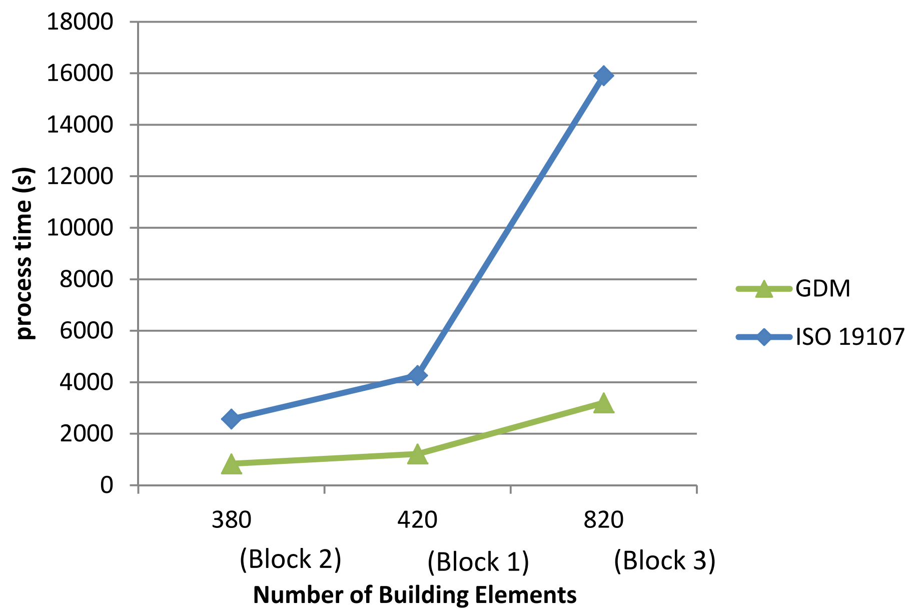


Figure 7.7: Processing time of topological relationship deduction using GDM and ISO19107

7.2.3.2. Configuration Generator and Constructability Analyzer

The possible configurations are then obtained using Configuration Generator in CoCoPS framework. This function is performed on each block type. The produced configurations are compared against the constructability criteria described in chapter 3 to reduce the list of possible configurations to feasible configurations.

The comparator function and similarity check are performed on all three block types at the same time to achieve a higher degree of standardization throughout the project. To this end, graph data models of each block type are combined to establish a single graph while, the configuration of building components in each block type is preserved. Thus a new function is developed to obtain the combined adjacency matrix by adding GDM of block types. Figure 7.8 shows the combined adjacency matrix obtained from two simple graphs.

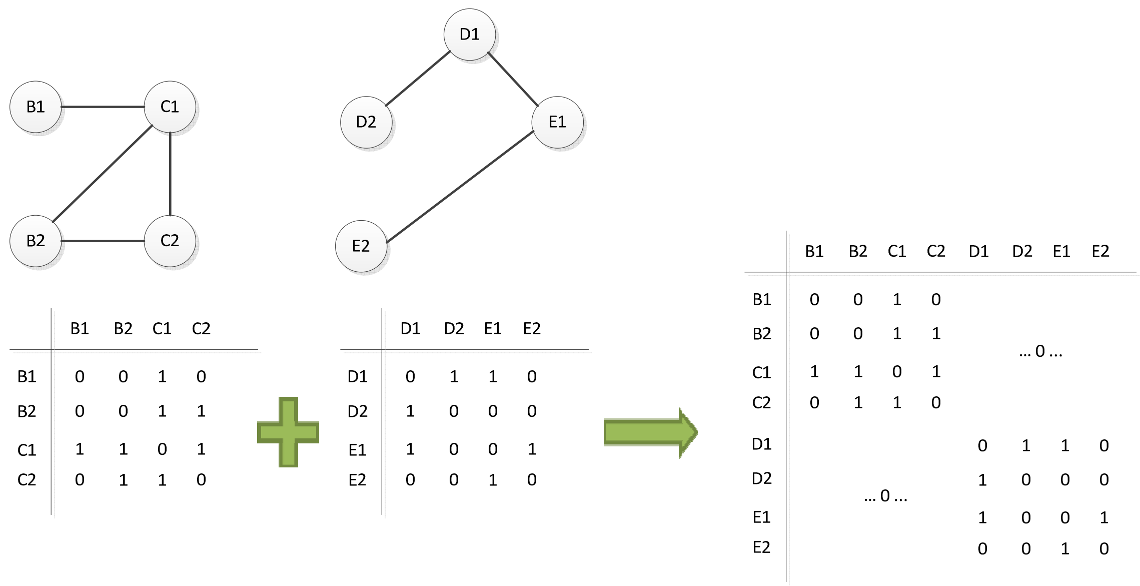


Figure 7.8: Combining graphs for generating configurations and similarity check

In the proposed "Comparator" function, a tabu search algorithm is utilized as a heuristic method to reduce computation time. In order to evaluate the efficiency of the developed tabu search, this method is compared against a combinatorial method. The combinatorial method adds an infeasible configuration to the black list accumulatively and a new configuration is checked with all records in the black list. The number of possible configurations for all block types is shown in Table 7.5. This table presents the number of feasible configurations and their respective processing time that are obtained using Tabu Search (TS) and combinatorial method. As described in Chapter 3, short-term function and tabu-time have been defined in such a way to prevent it from becoming trapped locally optimal solution. Thus using TS method, CoCoPS does not miss good configurations. The size of graph for each block type is demonstrated here to validate the cardinality of the set of feasible configurations and also to validate the performance of the employed TS method.

Table 7.5: The number of possible and feasible configurations using TS and combinatorial methods

	TS			Combinatorial		
Block Type	Number of possible configurations	number of feasible configurations	Process Time (hrs.)	Number of possible configurations	number of feasible configurations	Process Time (hrs.)
1	863,654	1,189	1.8	4,523,569	1,255	83
2	768,457	735	1.03	3,004,888	871	69
3	1,836,225	2,365	2.36	-	-	-
Combined Graph	45,888,103	4,156	4.09	-	-	-

The combinatorial method is not able to find the set of feasible solution for Block 3 and combined graph due to the excessive computation time required. Although the combinatorial method provides the exact number of feasible solutions, the processing time exponentially increase with increasing number of building elements (nodes) and topological relationships (edges). The small problems (less than 50 building elements) are easy to solve using the combinatorial method, but larger problems are more difficult to solve on a personal computer (due to limitation of data-segment amount imposed by Windows operating system). This table also indicates that performing the combinatorial method for the block type “3” and the “Combined Graph” is impossible on mini-workstation due to the size of graphs. Note that in the proposed TS algorithm the parameters settings are designed to be adjusted as a function of problem size.

The last function in Constructability Analyzer module is finding similar building components over the whole project for each feasible configuration. The sub-graph isomorphism algorithm which was developed in the CoCoPS framework is implemented on the combined graph.

In order to validate the performance of the selected algorithm for sub-graph isomorphism and the performance of implementation in .NET platform, the outcome of the function is compared with the benchmark problem that is reported by (Jos´e Luis L´opez Presa). The tests have been carried out on the same computer configuration described earlier in this chapter. All the programs have been compiled with the same compiler, C# compiler, and using the same optimization options. The real time (not CPU time) is considered as the execution time in all runs. The loading graph is excluded because of different graph size. A 30000 seconds CPU time limit is set for each program run. If a program is unable to finish by this time limit, no more tests for that or bigger size were performed for that program, and all previous results for that graph size and program were discarded.

The results are compared with the following algorithm reported in the literature: 1) Nauty 2.2. by (McKay, 1981), 2) Sinauto by (Presa, 2009), and Vf2 created by (Cordella et al., 2004). Since this study extends the Vf2 algorithm to carry weights on both edges and vertices, the original algorithm is called Original Vf2, and the extended algorithm is termed as Modified Vf2. The generated graphs for benchmark analysis are undirected and they are in the category of random graphs (Foggia et al., 2001). The results are shown in Figure 7.9.

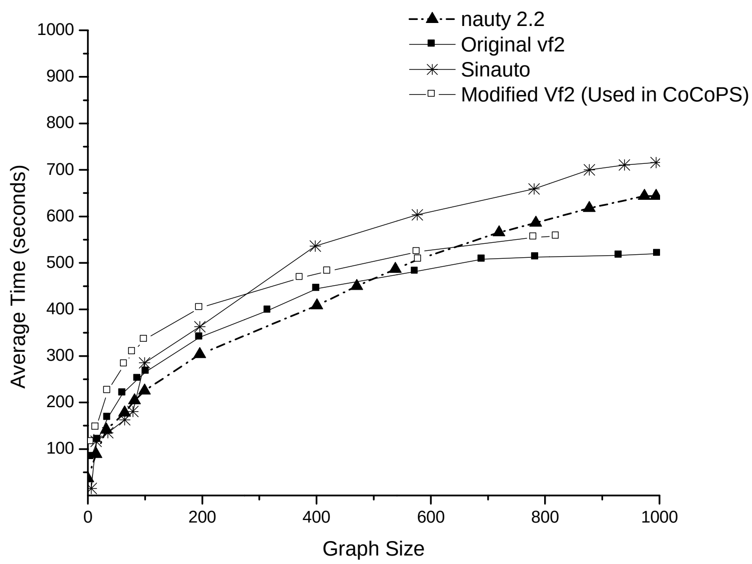


Figure 7.9: Average time of finding similar components vs. graph size

The processing time of similarity check for GDM of Diamond Town is shown by hollow-square label in the chart. Graph size for Diamond Town varies by block type. The result shows that the modified vf2 algorithm follows the same pattern as the reported algorithms. However, the average time of modified vf2 is slightly more than the original vf2 because of checking weight on edges. Moreover, the selected algorithm is the best fitted algorithm for attributed (with weights) sub-graph isomorphism. Because the computation time of the modified vf2 does not increase significantly by increasing the graph size. The time complexity of the utilized algorithm in the best case is $O(N^2)$ and in the worst case is $O(N!N)$.

7.3. Resource Optimization

The last module in the CoCoPS framework is optimizer which finds the best prefabrication configuration in terms of resource utilization. The objective function includes three constituents of project costs, namely: mould utilization cost, transportation cost, and installation cost. The mould utilization and transportation are obtained from the models described in chapter 5 and 6 respectively. The installation cost of components depends on following parameters: complexity of components, dimension, weight, number of lifting points and in-situ joints. The installation cost is calculated based on local regulations and aforementioned parameters (OSHS, 2002; Toffolon, 1984). With the shape and semantic information of each component from GDM these parameters are determined empirically using the local database for prefabricated projects.

In order to obtain the optimal configuration of components for prefabrication of the whole project; optimization models should be performed on all feasible configurations (4156 configurations in this project). However, it is extremely costly to apply the MILP model described in Chapter 5 on every feasible configuration. Therefore, to simplify the procedure and come up with a practical solution for large-scale projects, a hierarchical approach to plan and optimize resources of off-site fabrication is presented that can be modeled as a two-stage process.

The first stage provides a broad plan for the production plant using the model that has been explained in Chapter 3. The broad planning model is independent from production recipe but it has a ballpark of all cost constituents. Thus it can be performed for all configurations and reduce the size of optimal solution set to a few low cost configurations. In the second stage, the detail planning and transportation optimization, and installation cost models are run for the set of solution that has been obtained from broad planning stage.

In this case study, the best five configurations (termed as Config1 to Config5) in terms of total cost obtained from the broad planning stage are analysed for detail optimization in the second stage. The results are compared against three heuristic configurations which are defined as benchmarks. The first benchmark configuration is conventional approach with elemental building elements that are produced individually. The other benchmarks attempt to demonstrate the main achievement of the main objective for developing CoCoPS framework. Accordingly, these benchmarks are defined as follows:

1- Conventional Configuration (individual prefabricated building elements) (BM1)

2- Configuration with the minimum number of component types (BM2)

3- Configuration with the minimum number of identical components (BM3)

For model implementation the CPLEX 12.2.0.2 in GAMS 23.6.5 is used by Dell Precision T5500 with Intel(R) Xeon (R) CPU X5650 @ 2.67 GHz Processor with 48 GB of RAM running Windows 7 Professional 64-bit operating system.

Table 7.6 shows the five best configurations obtained from broad planning model in terms of total cost, production, transportation, and installation cost. This table also presents the results for benchmark configurations (BM1-BM3).

Table 7.6: Summary of broad planning results

	Mould Utilization Cost ($)	Transportation Cost ($)	Lifting Cost ($)	Total Cost ($)
BM1	$4,619,200	$728,400	$5,907,000	$11,254,600
BM2	$4,204,600	$795,600	$4,973,700	$9,973,900
BM3	$3,872,400	$881,600	$4,487,000	$9,241,000
Config 1	$3,936,900	$845,600	$5,266,500	$10,049,000
Config 2	$3,769,300	$782,300	$4,314,400	$8,866,000
Config 3	$4,126,900	$742,700	$4,189,100	$9,058,700
Config 4	$4,089,700	$1,033,900	$5,105,500	$10,229,100
Config 5	$3,957,000	$901,300	$4,948,400	$9,806,700

The results highlight that the traditional configuration is the most expensive configuration in terms of mould fabrication, and lifting cost; however, it is the cheapest one in terms of transportation cost. The latter is expected because of cuboid prefabricated components in traditional prefabrication method. The optimal configuration of components for this project is Config2 with a total cost of $8,866,000. Config2 is the cheapest configuration for mould fabrication and it is the second lowest lifting cost.

7.3.1. Resource based Production planning

The second optimization stage comprises three separate models for production planning, transportation and installation. The following describes the detail implementation of each model including the general assumptions, results and discussion.

7.3.1.1. General Assumptions

The construction cycle for this project is defined as production and installation of total required components for 80 flats. With this assumption the total number of construction cycles (T) is 55. According to the manufacturer capabilities for this particular project it is assumed that the required components for each construction cycle is produced within 25 working days (K=25) in a dedicated prefabrication plant. The production area for this project is limited to 10,000 m^2. The production plant is equipped with a concrete batching plant with a capacity of 350 m^3/day. The mould life cycle is estimated as 400 times. Mould fabrication cost is adjusted with the local market price and calculated for the existing mould types in each configuration as described in Chapter 3. The changeover penalty is assumed in the range of \$100 to \$400 per changeover for simple to complex moulds. The mould waste penalty is \$50. This is to distinguish solutions with the same total cost (not including waste penalty).

The number of component type (*j*), mould type (*i*), list of component group (*h*), mould adaptability ($MA_{i,h}$), and components type per group ($CTG_{h,j}$) which varies by configuration are obtained from CoCoPS for each configuration. The mould space and component volume are calculated according to the dimension of components which is stored in the database. Production recipe ($D_{j,k,t}$) is provided for the selected configurations based on the schedule demand.

7.3.1.2. Results and Discussion

Total mould utilization cost for the obtained configurations and benchmarks are presented in Table 7.7. In this table BM1 to BM3 are the heuristic configurations and Config1 to Config5 are the low cost configurations that are obtained from the broad planning stage. Cost advantage shows the comparison to BM1 (conventional elemental prefabrication). The maximum and minimum costs are presented by red and blue labels respectively. The optimal solution in terms of mould fabrication and utilization cost is achieved with total cost of $4,026,000 for Config 2, which is the configurations that obtained the lowest production cost in the first stage model as well. The production cost of Config 2 in the second stage is 6% higher than the first stage. This is because of demand list and delivery dates that have been applied in the second stage. For example, requiring more components on early days of construction cycle forces the model to employ more moulds so that the initial cost is increased.

The total cost of production for the conventional method (BM1) in this project is $4,387,000 comprising Mould Fabrication cost of $2,985,000, Mould Usage cost of $1,374,000, Changeover penalty of $28,000, and zero waste penalty. The idea of component grouping has been applied on traditional prefabrication method so that there is a changeover cost for bigger moulds that produces smaller components (for example a long beam mould produces shorter beams). Since these moulds are not very complicated and will be fully adjusted for prefabrication of smaller components, there is no waste penalty cost in this method. As can be expected, the idea of component grouping reduces the production cost of BM1 ($4,387,000) by 6% in comparison with the broad planning model ($4,619,200).

In the optimal configuration (Config2) the total cost is reduced by 9.22% in comparison with BM1. Although the Mould fabrication cost of the optimal configuration (Config2) is higher than the conventional method (BM1) because of complexity of moulds, the mould usage cost is lower than BM1 which indicates that the moulds are exploited to produce wider range of smaller components using the concept of component grouping. Further to this, low changeover and waste penalty costs show that the adopted moulds are almost fully utilized during the production.

Table 7.7: Results of mould utilization for Diamond Town project

	Total Cost ($)	Cost advantage to BM1 (%)	Mould Fabrication Cost ($)	Mould Usage Cost ($)	Changeover penalty ($)	Waste Penalty ($)
BM1	$4,387,000		$2,985,000	$1,374,000	$28,000	$0
BM2	$4,321,000	1.76	$3,265,000	$1,005,600	$42,000	$8,400
BM3	$4,236,000	3.80	$3,198,500	$971,700	$53,000	$12,800
Config 1	$4,183,000	5.12	$2,880,000	$1,290,800	$8,200	$4,000
Config 2	$4,026,000	9.22	$3,156,000	$864,000	$1,200	$4,800
Config 3	$4,298,000	2.30	$3,110,800	$1,171,200	$800	$15,200
Config 4	$4,222,500	4.13	$3,023,400	$1,185,750	$3,600	$9,750
Config 5	$4,098,500	7.28	$2,735,000	$1,352,050	$6,800	$4,650

7.3.2. Transportation Optimization (Containerization)

7.3.2.1. General Assumptions

The developed CLA model is utilized to determine the minimum number of trucks required for transportation of produced components to construction site. GA experiments are performed with the same crossover rates (0.6- 0.8) and mutation rates (0.2-0.4) as illustrated in Chapter 6. Population sizes are set up to five times the number of input components. One hundred generations are produced for each test run.

The problem is modeled for real truck situation. The truck dimension is 5 by 4 by 12 meters for width, height and length respectively. The maximum loading capacity of the trucks is 25 tons. The transportation cost of prefabricated components is adjusted to the local price and it is assumed as $50/truck per trip. The constant weight distribution is selected as packing strategy. The cost of transportation is shown in Table 7.8.

Table 7.8: Results of transportation cost for Diamond project

	Average Weight per truck (tons)	Transportation Cost ($)	Cost advantage to random placement (%)	Cost advantage to BM1 (%)
BM1	22.31	555,800	21.56	
BM2	16.85	735,850	27.89	-24.47
BM3	17.50	708,550	24.27	-21.56
Config 1	18.10	685,050	36.33	-18.87
Config 2	18.88	656,750	30.8	-15.37
Config 3	20.30	610,800	37.1	-9
Config 4	14.25	870,100	32.71	-36.12
Config 5	19.37	640,150	25.69	-24.91

The obtained results indicate that the first benchmark (BM1) which represents the traditional prefabrication has the minimum transportation cost. The result of detail transportation model is consistent with the broad planning model. However, the result of broad planning model is about 8% to 30% higher than the result of CLP model. Since the broad planning model for transportation is an empirical model, it can be assumed that it is a random sequence of components for component loading.

The transportation cost of traditional prefabrication method is $555,800 with the average truck weight of 21.31 tons. As can be expected, the traditional method achieved the maximum average weight and the minimum cost because of cuboid shape of components among all configurations. The other configurations have a higher transportation cost because of more complex building components. The “Config 3” has the minimum transportation cost among the selected configurations obtained from the broad planning stage. Config2, which has the minimum mould utilization cost stands in the third lowest transportation cost rank.

To represent the importance of containerization analysis, the model is run for random sequence of components rather than using GA. This scenario highlights the amount of cost saving that can be obtained for transportation if an optimal packing is followed. The transportation cost is compared with average of 100 random sequences for every selected configurations and benchmarks. As can be seen in Table 7.8, the cost saving for traditional method of prefabrication (BM1) is 21.56% because of cuboid components. However, the minimum saving for the configuration with higher degree of prefabrication is 25.69% is and the maximum savings is 36.33%. This indicates that although the transportation cost of new prefabrication method is slightly higher than the traditional method, the optimization model may save the transportation cost up to 36%.

7.3.3. The Installation Cost Analysis

7.3.3.1. General assumptions

The installation cost is calculated based on installation base rate. The base rate is determined for traditional prefabrication method from contractors' past experiences and their capabilities. The base rate comprises five installation processes namely: lifting set-up, safety check, crane movement, fitting, and installation (Equation 7.1). The base rate is adjusted for complex component types (i) using ShapeFactor(i). The Shapefactor of each component type (i) is a function of dimension, geometry, weight, number of lifting points (hoists) and in-situ joints or fixing points (see Equation 7.2). According to the designer and contractor database, the Shapefactor is calculated empirically and forms a library of preferred shapes for this project.

For example, the u-shape component shown in Figure 7.10 (left) comprises 11 smaller components which are commonly used in conventional approach. Installation of the u-shape component (left) is complicated, time consuming and costly. But it is obvious that installation time and cost of the combined prefabricated unit is cheaper than installation of 11 individual components.

$$BaseRate = f(\text{set-up,safety check,crane move,fitting,installation}) \tag{7.1}$$

$$Shapefactor(i) = g(\text{dimension, geometry, joints, fixing points, hoists}) \tag{7.2}$$

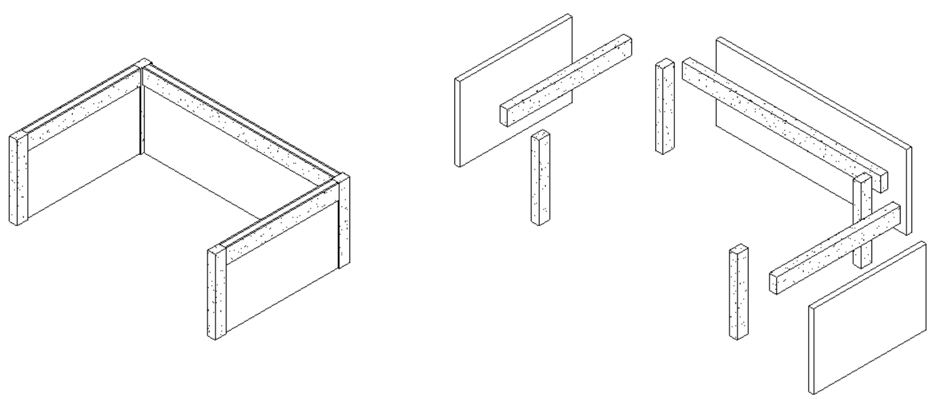


Figure 7.10: left: prefabricated unit, right: individual prefabricated elements

Having the BaseRate, Shapefactor(i) and the total number of each component type (DD_i), the total lifting cost is calculated using Equation 7.3.

$$Total\ Lifting\ Cost = \sum_{i=1}^{I} (BaseRate \times Shapefactor(i) \times DD_i) \tag{7.3}$$

7.3.3.2. Results and Discussion

In this project, the local construction team comprising 4 workers and one crane are able to install 12 elements in four hours. Having the geometric and semantic information of component types for each benchmark and selected configurations that obtained from constructability analyzer, the Shapefactor is empirically determined. Prices are adjusted to the local market of the project location. The total installation cost for this project is shown in Table 7.9.

Table 7.9: Results of installation cost for Diamond project

	Installation Cost ($)	Broad Planning Result ($)
BM1	**5,505,500**	**6,166,000**
BM2	4,542,400	4,315,200
BM3	4,302,400	3,936,600
Config 1	5,145,000	4,821,800
Config 2	**4,398,000**	**4,750,800**
Config 3	4,931,250	4,142,200
Config 4	4,994,500	4,880,200
Config 5	5,024,000	5,325,500

As can be seen in Table 7.9, the installation cost of all configurations including heuristic benchmarks are less than the traditional prefabrication method. It is mainly because of the number of prefabricated components in each configuration. For example, the traditional prefabrication method (BM1) consists of 1,101,100 building elements; however, the BM2 which represents the configuration with minimum number of identical components comprises 537,800 elements. Therefore the installation cost of BM2 is reduced by 28%. The obtained configuration which has the minimum mould utilization cost (Config2) has 614,200 building components. The installation cost of Config2 is reduced by 25.2% in comparison with the BM1. The obtained results from the broad planning stage also identified the BM1 as the most expensive solution in terms of lifting cost. However, with broad planning model, Config3 was identified as the optimal solution. The accuracy of broad planning optimization model for lifting cost is +-15% when it is compared with the detail planning stage.

7.4. Summary of resource utilization cost

The total cost of prefabrication of the "Diamond" project is calculated by adding "Mould Utilization", "Transportation" and "Installation" cost constituents. The total cost is demonstrated in Table 7.10.

Table 7.10: Summary of cost constituents for prefabrication of Diamond project

	Mould Utilization Cost ($)	Transportation Cost ($)	Lifting Cost ($)	Total Cost ($)
BM1	4,397,000	555,800	5,505,500	10,358,300
BM2	4,321,000	735,850	4,542,400	9,599,250
BM3	4,236,000	708,550	4,302,400	9,246,950
Config 1	4,183,000	685,050	5,145,000	10,013,050
Config 2	4,026,000	656,750	4,398,000	9,080,750
Config 3	4,298,000	610,800	4,931,250	10,140,050
Config 4	4,222,500	870,100	4,994,500	10,087,100
Config 5	4,098,500	740,150	5,024,000	9,862,650

The minimum and maximum of each cost constituents are highlighted by blue and red colors respectively. As can be seen in Table 7.10, the total cost of "Config 2" has the minimum total cost ($9,080,750). This table also indicates that the traditional prefabrication method has the maximum total cost ($10,358,300) among all heuristics and broad planning optimal configurations. The broad planning optimization model also identified the Config2 as the optimal solution with 2.4% lower cost than the cost of Config2 in detail planning model. The traditional configuration is identified as the worst configuration with 8.6% higher cost than the cost of traditional configuration that obtained from detail planning model. The result of detail optimization models reveals that the broad planning model is sufficiently reliable for a preliminary stage planning with 9% variance in total cost results.

Table 7.11 summarizes the cost advantage of the configurations in comparison with the conventional elemental approach for the various costs constituents. In this table negative numbers indicates cost reduction and positive numbers indicate cost increase. The mould utilization cost including mould fabrication and mould usage can be reduced by as much as 9.3% when a higher degree of prefabrication is applied. Although the transportation cost of complex configurations is 9%-36.1% more expensive than the traditional prefabrication method, the proposed model reduces the average transportation cost (for all configurations) by as much as 30% shown in earlier section. The installation cost of higher degrees of prefabrication is up to 15% cheaper than the traditional method. Finally, the economic advantage of the generated optimal configurations can be as much as 15%. The optimal configuration (Config 2) incurs less mould utilization and installation cost; however transportation cost is higher due to complex shape of components. The findings from the case study show that higher degree of prefabrication is economically viable.

Table 7.11: Cost advantage of configurations for all cost constituents

	Mould Utilization (%)	Transportation (%)	Installation (%)	Total (%)
BM2	1.76	-24.47	21.20	8.95
BM3	3.80	-21.56	27.96	13.10
Config 1	5.12	-18.87	7.01	4.45
Config 2	9.22	-15.37	25.18	15.17
Config 3	2.30	-9.00	11.65	6.28
Config 4	4.13	-36.12	10.23	3.68
Config 5	7.28	-24.91	9.58	6.04

7.5. Concluding Remarks

In this chapter, the CoCoPS framework is validated against a large-scale project. The project is a residential complex comprising 91 blocks of 10-storey buildings in four types with different floor plans. This case study illustrates that the CoCoPS framework is able to employ a higher degree of prefabrication for a large-scale project.

In order to implement the developed framework on large-scale projects, some of the functions should be tuned to reduce computation cost. The first function that is required to be adjusted is configuration generator in which, GDM of residential blocks are combined to obtain an adjacency matrix for the whole project. Therefore the Comparator function searches for similar components through the whole project to achieve a higher degree of standardization instead of finding similar components for each individual block type. The second function that is modified is optimizer in which a two-stage resource optimization method is employed in order to reduce the set of optimal solutions.

The first stage provides a broad plan for the production plant using the model that has been explained in Chapter 3. The broad planning model is independent from production recipe but it has a ballpark of all cost constituents. Thus it can be performed for all configurations and reduce the size of optimal solution set to a few low cost configurations. Although the broad plan model may not find the optimal solution, it is practical for large scale projects and it reduces the computation time significantly.

In this case study, the five best configurations in terms of mould utilization, transportation and installation cost were obtained from the first stage resource optimization (Config1 to Config5). The broad planning model also was run for three benchmarks (BM1 to BM3). The outcome of broad planning optimization was passed on to detail optimization for detail analysis.

In this project, 15% reduction in total resource utilization was achieved for Config2 using component grouping concept. The mould utilization, transportation and installation cost are resources that were considered in this study. The obtained results indicate that mould utilization cost of solution set (Config1 to Config5) and benchmarks BM2 and BM3 are less than the traditional prefabrication method (BM1). The cheapest configuration in terms of mould utilization is Config2 which reduces the mould utilization cost by 9%.

The container loading optimization model shows that the transportation cost of the traditional approach (BM1) is the cheapest configuration (because of cuboid components). However, transportation cost of BM1 could not change the optimal solution in terms of the total prefabrication cost. The project optimal configuration (Config2) incurs 15% more transportation cost in comparison to BM1. This study also showed that using an optimization model for loading trucks in this project reduces the transportation cost by up to 30% when compared with random placing.

An empirical model for installation cost analysis is proposed in this chapter. This model shows that the installation cost of high-level prefabrication is reduced by up to 28%. BM3 which has the minimum number of identical component is the optimal solution in terms of installation cost.

CHAPTER 8: RESEARCH CONCLUSIONS AND FUTURE WORKS

8.1. Conclusions

The main purpose of this study is to develop a framework to extend the traditional prefabrication of individual building elements in IBS to a higher degree of prefabrication to reduce project cost by providing a new level of standardization. This framework requires enhanced coordination between designer and producer to consider physical, functional and spatial properties of building elements. Furthermore, this framework must give maximum flexibility to designer while considering standardization of products, and constructability criteria.

In this regard, the proposed framework which is called Component Configuration of Prefabricated Structures (CoCoPS) configures the grouping of elements into higher level components in order to minimize the total number of components so as to reduce transportation and installation costs and at the same time to maximize the number of identical components for reason of economy of scale in terms of mould fabrication cost.

The framework extracts semantic, geometric, and geometrical properties of building elements from an intelligent parametric model. Topological relationships among building elements are deduced so that all possible configurations of building elements are generated. A rule-based database is utilized to check constructability of the generated configuration in terms of design, production, transportation, and installation. Two optimization models are proposed and implemented on the set of feasible configurations to evaluate them and obtain the most economic configuration of components in term of resources required for production and transportation. The CoCoPS prototype is implemented on a case study to verify the research findings. To achieve this goal, the following research components have been developed:

1- Development of Component Configuration of Prefabricated Structures (CoCoPS) framework as a new approach to industrialized building system for integration between design-production coordination, standardization and prefabrication. CoCoPS adopts the idea of grouping building elements to obtain high level components. Essentially, the framework was designed to extract topological relationships and geometrical properties of building elements from IFC file to be represented in a Graph Data Model (GDM) sequentially supplemented with a Geometric Network Model (GNM) from which a novel configuration string "S" was built. This string forms the basis by which all possible configurations can be generated using the Depth First Search (DFS) graph traversing algorithm. Feasibility was obtained by comparison of sub-graphs within the configuration string against production and constructability criteria and the library of preferred shapes using a modified sub-graph isomorphism algorithm. A tabu search algorithm was applied to reduce computational time in finding the set of feasible configurations. The graph isomorphism algorithm was also utilized to optimize the mould types by finding similar components in the sub-graphs within the configuration string.
2- To represent topological relationships a novel semantic Graph Data Model (GDM) was developed. The GDM is able to handle various queries and complex network analysis on building elements within a building. The GDM was derived using weighted graph elements and formulated with graph-theory and adjacency matrix

notations. Thus it simplifies the abstraction of the topological relationships among 3D objects using the node-edge structure of the graph. The GDM was enriched by adding semantic information as weights to nodes and edges using IFC capabilities. The elements of the proposed GDM make it an elaborated intelligent model to represent topological relationships that will be able to handle wide ranges of queries efficiently. For example, it would be carry on DFS algorithm on weighted graph about 15% faster than Sinauto and Nauty 2.2.

Having a weighted graph data structure, complex topological queries can be implemented through advanced graph algorithms such as Sub-graph isomorphism. Moreover SDT makes the GDM a knowledge embedded model which would be able to run rule-based queries and constraints such as weighted graph isomorphism algorithm required in query 2 in Chapter 4. The GDM presented in this study contributes to the advancement of research in the area of 3D topological models for AEC applications and also overcomes several limitations of the existing models in following ways: first, it explicitly represents the elements (structural and non-structural) of buildings using a weighted graph data structure; second, since the proposed model is not limited to any specific geometric representation of 3D objects such as B-rep, SweptSolid, or CSG a wide range of 3D objects can be modeled. However, the proposed model does not cover curved-shape building elements. Using IFC as a data exchange platform enables the GDM to exploit the pre-defined topological relationships in IFC to significantly reduce the deduction time. Fourth, previous graph data models such as those developed by Borrmann (2009), Van Treeck (2009) and Lee and Kwan (2005) are limited to adjacency and connectivity for the sake of specific purpose of finding shortest path among spaces in a building, whereas the GDM covers all four major spatial relationships for all AEC/FM applications.

3- In order to automatically generate the GDM, a topology-driven algorithm was required to extract such topological relationships among building elements. An IFC-based deduction algorithm was developed for four major types of topological information comprising: Connectivity, Containment, Separation, and Intersection. Using geometric/topologic representation in IFC data model, the developed IFC-based algorithm enhanced the deduction performance in three ways. First, IFC files were used to directly extract the topological primitives and their geometric information instead of obtaining from lines and points in a CAD environment. Second, the SweptSolid, CSG, and B-rep representations of 3D objects which are supported by IFC were used to represent a wider range of 3D objects when compared to the conventional CAD modeling. Third, IFC comprises predefined relationships between the building objects, eliminating the need for pairwise comparison to deduce binary relationships. These entities were used to reduce computation time and complexity. Indeed using IFC as a universal standard platform facilitates data exchange between different professionals so that the algorithm is independent of software.

4- In order to find the optimal degree of prefabrication in project, a two-stage optimization model was proposed. This model was aimed to minimize the resource utilization of each configuration. The mould fabrication and mould utilization, transportation and installation were detected as the most important resources of prefabrication. The first stage of optimization model tries to reduce optimal solution domain, while, the second stage leads to an optimal detail production plan and container loading plan. The installation cost in both stages is based on empirical cost factors that are obtained from local markets. These prices are then adjusted for component types using shape factor. The mould utilization and transportation optimization models are as follows:

a. Mould fabrication was identified as the most expensive resource in prefabrication plants which have certain capacity and life cycle. In this regard, the first optimization model is meant to maximize mould usage in its life cycle to produce a range of building components. In order to account for component complexity and capability of moulds producing different building components, two new concepts were defined which are Component Group and Mould Adaptability. In this model, the objective function is minimizing mould fabrication, mould utilization, changeover, and mould waste cost, while considering mould capability constraints, factory constraints, and installation demand constraints. A Mixed Integer Linear Programming (MILP) model was created to formulate the problem. The proposed MILP model accommodates and presents novel treatment for key aspects of precast production planning resource and activity such as sequence-dependent changeovers, mould allocation, delivery dates, etc, and gives the exact number of moulds required for planning horizon and schedule for each mould. The developed model could significantly reduce mould fabrication and mould usage cost, so that the conventional method in prefabrication is not the optimal degree of prefabrication in projects.

b. The second costly resource is transportation. The transportation of irregular three-dimensional components which are produced with a higher degree of prefabrication will be extremely costly if randomly placed in trucks. Therefore the issue of transportation was studied in this research as a containerization problem. A novel approach for loading irregular 3D objects into trucks was proposed in this study to optimize transportation of prefabricated components. The transportation problem was modeled as a particular form of general Container Loading Problem (CLP). In this model the best sequence of locating 3D irregular-shaped objects that are not restricted to any shapes or orientations into 3D rectangular containers was obtained. The stability of components and loading capacity (weight) are the problem constraints. An automated system was developed to combine a heuristic sliding algorithm and GA sequence generator. The sliding algorithm developed in this study uses voxels (3D pixels) to represent non-standard irregular 3D objects and place them into the containers. Using voxels helps the sliding algorithm to overcome any orientation and size restrictions. A genetic algorithm is used to generate sequences of the given objects to be allocated. The evaluation criterion of loading is the unused space percentage of each allocation considering the packing strategy. The case study result indicated that the proposed model can reduce the transportation cost by 30% in comparison with the random loading scenario.

The abovementioned research components had been implemented in a prototype CoCoPS that is applied on a case study. This case study is the construction of a residential complex comprising 91 blocks of 10- storeys with different floor plans.

The results of the case study demonstrated that the proposed CoCoPS framework can successfully integrates the described research components to add more value to the project. This value was added to the project through adopting a higher degree of prefabrication. The framework also gave the maximum flexibility to the designer to not follow the closed-systems in IBS.

The case study also proved the functionality of the algorithms (e.g. sub-graph isomorphism, tabu search, topological relationship deduction) that were employed in CoCoPS for large scale projects.

The results of case study indicated that the adoption of a higher degree of prefabrication reduces the total resource utilization cost by 14%. The optimal solution has 7% and 25% cost advantage in terms of mould utilization and installation respectively. However, this solution is 15% more expensive than the traditional prefabrication method in transportation which is due to the complex and irregular shape components that were employed.

The mould optimization model provided the optimal production plan in which the minimum changeover occurred which means that the moulds were fully utilized during the production. Further to this, the waste penalty cost in the objective function indicated that the idea of component grouping was successfully implemented so that the moulds are employed to produce a variety of building component types at the same time.

8.2. Limitation and Future works

This research had spearheaded a worthwhile direction in adopting a higher level of prefabrication in construction and developed a novel approach in integration of production, transportation and installation in off-site fabrication for more cost advantages in projects, but more can still be done to build on the foundation provided by this research to further improve the performance of industrialized building systems. The following limitations and/or recommendations are noted and discussed.

The proposed CoCoPS framework was developed to obtain a higher degree of prefabrication in off-site construction. This framework is not limited to any specific construction system, material, or structural design. However, all most all of the case studies were focused on concrete structures because it is common in construction projects. The concrete structures limited some of the models in CoCoPS framework because of its weight and structural behavior. However, in some countries, timber, wood and steel structures are employed in construction projects. Further analysis is required on various structural systems and construction methods to verify the functionality of the proposed framework and the potential savings that can be afforded.

The constructability constraints that are considered in constructability analyzer were limited to shape, size and weight. Therefore, some of the configurations may not be practically feasible. In this regard, a library of preferred components was added to the framework to overcome this issue. However, a further study in this area can automatically consider design criteria to detect infeasible configurations. This feature may bring more flexibility to designers and may result in further reduction in resource utilization cost.

The installation cost was determined based on installation base rate. The base rate is determined for traditional prefabrication method (individual beam, column, wall or slab) from contractor past experiences and their capabilities. The base rate comprises of five installation processes namely: lifting set-up, safety check, crane move, fitting, and installation. The base rate was adjusted for more complex component types using a shape factor parameter empirically. Although the obtained results were satisfactory, developing a knowledge base or mathematical model may incorporate more real factors in installation so that more precise results may be obtained.

Although the proposed model for container loading in this study has obtained satisfactory results; the proposed approach can be improved by considering more real constraints (such as load bearing capacity of underneath objects), installation sequence, and packing strategy in model.

The scope of this research for representation of topological relationships of building elements is limited to boundary representation and SweptSolid. However, other advance topological representation can be considered for further improvement of the GDM. Moreover, since the GDM is constructed upon graph theory, it is capable of handling more advance queries. Further research area may exploit advance graph algorithms to run topological queries on GDM.

In order to obtain the optimal configuration of prefabricated elements, the present research focuses on resource utilization including mould fabrication and usage cost, transportation (containerization) and installation cost. Although these objectives represent the respective concerns of the project owner, all project parties may consider other evaluation criteria in reality.

Finally, a friendly user interface would be beneficial to the application of the framework in the industry. The 3D model viewer can facilitate the users in modification of input data, evaluation of feasible configurations, and location of components within trucks. The output interface can present an automatic generated gantt chart of production plan, the list of components and their 3D view for number of different configurations for easy reference.

References

Al-Hussein, M., Alkass, S., & Moselhi, O. (2005). Optimization algorithm for selection and on site location of mobile cranes. *Journal of Construction Engineering and Management, 131*, 579-590.

ASCE. (2011). Prefabrication and modularization Increasing Productivity in the construction Industry. In.

Balbontin-Bravo, E. (1998). Simulation of large precast operations. In *Winter Simulation Conference Proceedings* (Vol. 2, pp. 1311-1317).

Banzhaf, W. (1990). The “molecular” traveling salesman. *Biological Cybernetics, 64*, 7-14.

Bischoff, E. (1991). Stability aspects of pallet loading. *Operations-Research-Spektrum, 13*, 189-197.

Bischoff, E. E., & Ratcliff, M. S. W. (1995). Issues in the development of approaches to container loading. *Omega, 23*, 377-390.

Blickle, T., & Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput., 4*, 361-394.

Bock, T. (2004). Construction robotics and automation: past-present-future. In *Automation Congress, 2004. Proceedings. World* (Vol. 15, pp. 287-294).

Borrmann, A., & Rank, E. (2009). Topological analysis of 3D building models using a spatial query language. *Advanced Engineering Informatics, 23*, 370-385.

Bortfeldt, A., & Gehring, H. (1998). Ein Tabu Search-Verfahren für Containerbeladeprobleme mit schwach heterogenem Kistenvorrat. *OR Spectrum, 20*, 237-250.

Bottom, D., Gann, D., Groa`k, S., & Meikle, J. (1994). *Innovation in Japanese Prefabricated House-Building Industries*. London: Construction Industry Research and Information

Association, London.

BSRIA. (1999). *Prefabrication and preassembly - applying the techniques to building engineering services*. Bracknell: BSRIA.

Bunke, H. (1999). Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 917-922.

Carpenter, H., & Dowsland, W. B. (1985). Practical Considerations of the Pallet-Loading Problem. *The Journal of the Operational Research Society, 36*, 489-497.

Chan, W., & Hu, H. (2002). Production scheduling for precast plants using a flow shop sequencing model. *Journal of Computing in Civil Engineering, 16*, 165.

Chan, W. H., & Lu, M. (2005). Logistics and operations simulation in precast viaduct construction: Case study. In *Proceedings of the 2005 ASCE International Conference on Computing in Civil Engineering* (pp. 1373-1384).

Chan, W. T., & Hu, H. (2002). Constraint programming approach to precast production scheduling. *Journal of Construction Engineering and Management, 128*, 513-521.

Cheong, G. K. (1997). Fully precast system at Choa Chu Kang. In (Vol. 8). Precastech Newsletter On Line.
Choi, J., & Lee, J. (2009). 3d geo-network for agent-based building evacuation simulation. *3D Geo-Information Sciences*, 283-299.
Chokri, K., & Mathieu, K. (2009). A Simplified Geometric And Topological Modeling Of 3D Building Enriched By Semantic Data: Combination Of Surface-Based And Solid-Based Representations. In *ASPRS 2009 Annual Conference Baltimore, Maryland*.
Christoph van Treeck, R. R., and Ernst Rank. (2003). SIMULATION BASED ON THE PRODUCT MODEL STANDARD IFC. In *Eighth International IBPSA Conference* (pp. 1293-1300). Eindhoven, Netherlands
CIB, r. (2010). New Perspective in Industrialisation in Construction - A State-of-the-Art. *CIB Publication 329*.
CIRIA. (1997). *Snapshot - Standardization and Pre-assembly*. London: Construction Industry Research and Information Association,.
CIRIA. (1999). *Adding Value to Construction Projects through Standardisation and Pre-assembly, Report R176*. London: Construction Industry Research and Information Association.
CIRIA. (2000). *Standardisation, Pre-assembly and Modularisation– A Client's Guide*. London: Construction Industry Research and Information Association, London.
Coors, V. (2003). 3D-GIS in networking environments. *Computers, Environment and Urban Systems, 27*, 345-357.
Cordella, L., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1367-1372.
Davis, L. (1985). Job shop scheduling with genetic algorithms. In *First Int. Conf. Genetic Al#orithms and their Applications,* (pp. 136-140). Erlbaum.
Dawood, N. N. (1995). Scheduling in the precast concrete industry using the simulation modelling approach. *Building and Environment, 30*, 197-207.
Dawood, N. N., & Neale, R. H. (1993). A capacity planning model for precast concrete building products. *Building and Environment, 28*, 81-95.
Dietz, A. G. H. (1946). *Dwelling house construction*: D. Van Nostrand.
Diez, R., Padrin, V. M., Abderrahim, M., & Balaguer, C. (2007). AUTMOD3: The integration of design and planning tools for automatic modular construction. *International Journal of Advanced Robotic Systems, 4*, 457-468.
Diez, R., Padrón, V. M., Abderrahim, M., & Balaguer, C. (2007). AUTMOD3: The integration of design and planning tools for automatic modular construction. *International Journal of Advanced Robotic Systems, 4*, 457-468.
Domínguez, B., García, Á., & Feito, F. (2011). Semantic and topological representation of building indoors: an overview.
Egan, J. (1998). *Rethinking construction*. London: HSMO.
Egeblad, J., Garavelli, C., Lisi, S., & Pisinger, D. (2010). Heuristics for container loading of furniture. *European Journal of Operational Research, 200*, 881-892.
Ellul, C., & Haklay, M. M. (2009). Using a B-Rep Structure to Query 9-Intersection Topological Relationships in 3D GIS – Reviewing the Approach and Improving

Performance 3D Geo-Information Sciences. In J. Lee & S. Zlatanova (Eds.), (pp. 127-151): Springer Berlin Heidelberg.

Fasano, G. (2004). A MIP approach for some practical packing problems: Balancing constraints and tetris-like items. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies, 2*, 161-174.

Fasano, G. (2008). MIP-based heuristic for non-standard 3D-packing problems. *4OR, 6*, 291-310.

Fogel, D. (1988). An evolutionary approach to the traveling salesman problem. *Biological Cybernetics, 60*, 139-144.

Fogel, D. (1990). A Parallel Processing Approach to a Multiple Traveling Salesman Problem Using Evolutionary Programming. In *Proceedings on the Fourth Annual Parallel Processing Symposium* (pp. 318–326). Fullterton, CA.

Foggia, P., Sansone, C., & Vento, M. (2001). A performance comparison of five algorithms for graph isomorphism. In *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition* (pp. 188-199).

Franz, G., Mallot, H., Wiener, J., & Neurowissenschaft, K. (2005). Graph-based models of space in architecture and cognitive science-a comparative analysis. In *Proceedings of the 17th International Conference on Systems Research, Informatics and Cybernetics (InterSymp '2005), Architecture, Engineering and Construction of Built Environments* (pp. 30-38).

Friedman A., C., V. (1997). Cost Reduction through Prefabrication: A Design Approach. *Housing and Society, 24*, 1-14.

G., B. A. W. (2012). Container Loading Problems-A State-of-the-Art Review.

GALLEGO, M., LAGUNA, M., MARTÍ, R., & DUARTE, A. (2011). Tabu Search with Strategic Oscillation for the Maximally Diverse Grouping Problem. *Journal of the Operational Research Society*.

Garey, M. R., & Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*: W. H. Freeman \\& Co.

Gehring, H., & Bortfeldt, A. (1997). A Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operational Research, 4*, 401-418.

Gehring, H., & Bortfeldt, A. (2002). A Parallel Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operational Research, 9*, 497-511.

Gibb, A. (2001). Pre-assembly in construction: A review of recent and current industry and research initiatives on pre-assembly in construction. *CRISP Consultancy Commission (00/19)*.

Gibb, A., & Isack, F. (2003). Re-engineering through pre-assembly: client expectations and drivers. *Building Research & Information, 31*, 146-160.

Gibb, A. G. F. (1999). *Off-site fabrication: prefabrication, pre-assembly and modularisation*: Wiley.

Gibb, A. G. F. (2001). Standardization and pre-assembly-distinguishing myth from reality using case study research. *Construction Management & Economics, 19*, 307-315.

Gibb, A. G. F. S., W. D. (1999). Technical and managerial challenges of facade retention. *Proceedings of the Institution of Civil Engineers: Structures and Buildings, 134*, 219-229.

Girmscheid, G. S., F. (Eds.). (2010). Context of Industrialisation. In *New Perspective in Industrialisation in Construction A State-of-the-Art Report*
(Vol. 1, pp. 3-13). Zurich, Eigenverlag des IBB an der ETH Z¨urich: CIB.

Glover, F., & Laguna, M. (1993). *Tabu search*: John Wiley & Sons, Inc.

Goldberg, D., & Bridges, C. (1990). An analysis of a reordering operator on a GA-hard problem. *Biological Cybernetics, 62*, 397-405.

Goodier, C., & Gibb, A. (2004). Barriers and opportunities for off-site production. *PROSPA, Loughborough University, Loughborough*.

Grabska, E., Łachwa, A., & Ślusarczyk, G. (2012). New visual languages supporting design of multi-storey buildings. *Advanced Engineering Informatics, 26*, 681-690.

Groák, S. (1992). *The Idea of Building,* . Routledge, London: E & FN Spon.

Hao, H. (2007). A study of resource planning for precast production. *Architectural Science Review, 50*, 106-114.

Hassim, S., Jaafar, M. S., & Sazalli, S. A. A. H. (2009). The Contractor Perception Towers Industrialised Building System Risk in Construction Projects in Malaysia. *American Journal of Applied Sciences, 6*, 937-942.

Helena Johnsson, L. M., Stefan Persson. (2007). ICT support for industrial production of houses : the Swedish case. In *24th W78 Conference Maribor 2007* (Vol. 1, pp. 407-413). Maribor: CIB.

Herbert, G. (1984). *In Gropius, W. and Wachsmann, K.(eds), The Dream of the Factory-made House*. Cambridge, MA: MIT Press.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*: University of Michigan Press.

Huang, K., Wu, S., & Wang, M. (2005). Study on the Storage and Transportation Optimization of Prefabrication Factory.

ISO 10303-42. (1994). Industrial automation systems and integration-product data representation and exchange. In *ISO*.

J. Grefenstette, R. G., B. Rosmaita and D. Van Gucht,. (1985). Genetic algorithms for the traveling salesman problem. In *First Int. Conf. Genetic Algorithms and their Applications* (pp. 160-168). Erlbaum.

Jaillon, L., & Poon, C. S. (2009). The evolution of prefabricated residential building systems in Hong Kong: A review of the public and the private sector. *Automation in Construction, 18*, 239-248.

Junid, S. M. S. (1986). Industrialised Building System. In *Proceedings of a UNESCO/FEISEAP Regional Workshop*. Malaysia UPM Serdang.

Kamar, K., Alshawi, M., & Hamid, Z. (2009). Barriers to Industrialized Building System (IBS): The Case of Malaysia. *BuHu 9th International Postgraduate Research Conference*, 471-484.

Kamar, K. A. M., Hamid, Z. A., Azman, M. N. A., & Ahamad, M. S. S. (2011). Industrialized Building System (IBS): Revisiting Issues of Definition and Classification. *International Journal of Emerging Sciences, 1*, 120.

Kuo-Chuan Shih, S.-S. L. a. C.-N. H. (2005). Precast Storage and Transportation Planning via Component Zoning Optimization. *22nd International Symposiom on Automation and Robotics in Construction ISARC 2005, Ferrera , Italy*.

Laat, R., & Berlo, L. (2011). Integration of BIM and GIS: The development of the CityGML GeoBIM extension. *Advances in 3D Geo-Information Sciences*, 211-225.

LaCourse, D. E. (1995). *Handbook of solid modeling*: McGraw-Hill.

Lam, K. P., Wong, N. H., Shen, L. J., Mahdavi, A., Leong, E., Solihin, W., Au, K. S., & Kang, Z. (2006). Mapping of industry building product model for detailed thermal simulation and analysis. *Advances in Engineering Software, 37*, 133-145.

Lamarche, F., & Donikian, S. (2004). Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. In *Computer Graphics Forum* (Vol. 23, pp. 509-518): Wiley Online Library.

Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. *Artificial Intelligence Review, 13*, 129-170.

Lee, J., & Kwan, M.-P. (2005). A combinatorial data model for representing topological relations among 3D geographical features in micro-spatial environments. *International Journal of Geographical Information Science, 19*, 1039 - 1056.

Lessing, J. (2006). *Industrialised house-building: concept and processes*: Department of Construction Sciences, Lund University.

Leu, S. S., & Hwang, S. T. (2002). GA-based resource-constrained flow-shop scheduling model for mixed precast production. *Automation in Construction, 11*, 439-452.

Linus Malmgren, P. J., Thomas Olofsson. (2011). Product modeling of configurable building systems – a case study. *Journal of Information Technology in Construction, 16*, 697-712.

M. S., J. Lin, K. K., A. Bahrami, G. Leininger, & F. Liou. (1991). Hybrid Expert System and Operations Research for Solving Nesting Problems. In *The World Congress on Expert Systems* (pp. 1223-1231). Orlando Florida.

Mack, D., Bortfeldt, A., & Gehring, H. (2004). A parallel hybrid local search algorithm for the container loading problem. *International Transactions in Operational Research, 11*, 511-533.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming, 10*, 147-175.

McKay, B. D. (1981). Practical graph isomorphism.

Messmer, B. (1995). Efficient graph matching algorithms. *Institute for Applied Computer Science and Mathematics, University of Bern, Bern, Switzerland*.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer.

Molenaar, M. (1990). A formal data structure for 3D vector maps. In *Proceedings of EGIS'90* (Vol. 2, pp. 770-781). Amsterdam, The Netherlands.

Montes, J., Camps, I., & Fúster, A. (2011). Industrialización en la vivienda social de Madrid. *Informes de la Construcción, 63*, 5-19.

Nawi, M. M. N., KAMAR, K. A. M., ABDULLAH, M. R., HARON, A. T., LEE, A., & ARIF, M. (2007). ENHANCEMENT OF CONSTRUCTABILITY CONCEPT:

AN EXPERIENCE IN OFFSITE MALAYSIA CONSTRUCTION INDUSTRY. In.

Nguyen, T.-H., Oloufa, A. A., & Nassar, K. (2005). Algorithms for automated deduction of topological information. *Automation in Construction, 14*, 59-70.

Nguyen, T. H., & Oloufa, A. A. (2001). Computer-generated building data: Topological information. *Journal of Computing in Civil Engineering, 15*, 268-274.

OSHS. (2002). Approved Code of Practice for The Safe Handling, Transportation and Erection of Precast Concrete. In *Transportation, Handling and Erection* (pp. 36-41). New Zealand: Occupational Safety and Health Service, Department of Labour.

Parreño, F., Alvarez-Valdes, R., Tamarit, J. M., & Oliveira, J. F. (2008). A Maximal-Space Algorithm for the Container Loading Problem. *INFORMS Journal on Computing, 20*, 412-422.

Paul, N., & Borrmann, A. (2009). Geometrical and topological approaches in building information modelling.

Pigot, S., & Hazelton, B. (1992). The fundamental of a topological model for a 4-D GIS. In (Vol. 2, pp. 580-590).

Pilouk, M. (1996). *Integrated modelling for 3D GIS*: Enschede: ITC.

Pisinger, D. (2002). Heuristics for the container loading problem. *European Journal of Operational Research, 141*, 382-392.

Plümer, L., & Gröger, G. (1996). Nested maps—a formal, provably correct object model for spatial aggregates. In *Proceedings of the 4th ACM international workshop on Advances in geographic information systems* (pp. 76-83): ACM.

Polat, G. (2010). Precast concrete systems in developing vs. industrialized countries. *Journal of Civil Engineering and Management, 16*, 85-94.

Poshyanonda, P. a. D., C. (1992). A hybrid approach to composite stock cutting: artificial neural networks and genetic algorithms. *Robotics and Manufacturing Recent Trends in Research Education and Applications, 2*.

Presa, J. L. L. (2009). *Efficient algorithms for graph isomorphism testing*. Ph. D. dissertation, Licenciado en Informatica Madrid.

Raper, J. (2000). *Multidimensional geographic information science*: CRC.

Retik, A., & Warszawski, A. (1994). Automated design of prefabricated building. *Building and Environment, 29*, 421-436.

Rikkers, R. (1993). A query oriented implementation of a 3d topologic data structure. *Proceedings of EGIS'93 Genoa, Italy, 2*, 1411-1420.

Rivard, H., Mora, R., & Bedard, C. (2000). Geometrical reasoning and the conceptual design of building structures. In *8th International Conference on Computing in Civil and Building Engineering, August 14, 2000 - August 16, 2000* (Vol. 279, pp. 177-184). Stanford, CA, United states: American Society of Civil Engineers.

Rogan, A., & Lawson, R. (2000). Bates-Brkljac 2000. *Value and Benefits Assessment of Modular*.

S. Dalla Costa, E. R., and M. Rumor. (2011). A CITYGML 3D GEODATABASE FOR BUILDINGS' ENERGY EFFICIENCY. In *28th Urban Data Management Symposium* (Vol. Volume XXXVIII, pp. 19-24). Delft, The Netherlands: ISPRS Archives.

Sarja, A. (1998a). *Open and Industrialised Building*. Routledge, London.: E & FN Spon

Sarja, A. (1998b). *Open and Industrialised Building International Council for Building Research, Studies Documentation*: E & FN Spon.

Schepers, S. P. F. a. J. (2004). A Combinatorial Characterization of Higher-Dimensional Orthogonal Packing. *Math. Oper. Res.*

Seokho, L. J. L. (2010). Efficient Topological Data Models For Spatial Queries in 3D Gis. In *Geospatial Data and Geovisualization: Environment, Security, and Society Special Joint Symposium of ISPRS Commission IV and AutoCarto* (Vol. Volume XXXVIII Part 4). Orlando, Florida, USA.

Suter, G., & Mahdavi, A. (2004). Elements of a representation framework for performance-based design. *Building and Environment, 39*, 969-988.

Syswerda, G. (1991). Schedule optimization using genetic algorithms. In L. Davis (Ed.), *A Handbook of Genetic Algorithms* (pp. 332-349). Amsterdam: Van Nostrand Reinhold.

Tatum, C. B., Vanegas, J. A., & Williams, J. M. (1987). Constructability improvement using prefabrication, preassembly and modularization. *Source Document No. 25*.

Thanoon, W., Peng, L. W., Kadir, M. R. A., Jaafar, M. S., & Salit, M. S. (2003). The Essential Characteristics of Industrialised Building System. In (Vol. 10, pp. 11).

Toffolon, R. L. (1984). Lifting device for massive precast concrete wall units. In: Google Patents.

Tse, R. O. C., & Gold, C. (2003). A proposed connectivity-based model for a 3-D cadastre. *Computers, Environment and Urban Systems, 27*, 427-445.

Ullmann, J. (1976). An algorithm for subgraph isomorphism. *Journal of the ACM (JACM), 23*, 42.

van Treeck, C., & Rank, E. (2007). Dimensional reduction of 3D building models using graph theory and its application in building energy simulation. *Engineering with Computers, 23*, 109-122.

Warszawski, A. (1984). Production planning in prefabrication plant. *Building and Environment, 19*, 139-147.

Warszawski, A. (1999). *Industrialized and automated building systems*: E & FN Spon.

Warszawski, A., & Ishai, E. (1982). Long range planning of prefabrication industry in a national economy (summary). *Building and Environment, 17*, 47-54.

Warszwsaki, A. (1990). *Industrialization and Robotics in Building Construction*: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Wei, G., Ping, Z., & C, J. U. N. (1998). *Proceedings of ISPRS Commission IV*.

Zhai, X., Tiong, R. L. K., Bjornsson, H. C., & Chua, D. K. H. (2008). Simulation-based planning for precast production with two critical resources. In *Proceedings - Winter Simulation Conference* (pp. 2519-2526).

Zlatanova, S. (2000). 3D GIS for urban development. *ITC dissertation series (Netherlands)*.

List of Publications

Journal Papers

- A. Khalili and D.K.H. Chua, "An IFC-based framework to move beyond individual building elements towards configuring higher level prefabrication", *Journal of Computing in Civil Engineering*, ASCE, 2011.
- A. Khalili and D.K.H. Chua, "A Graph Data Model for topological representation of building elements using IFC", Accepted by *Journal of Computing in Civil Engineering*, ASCE, 2012.
- A. Khalili and D.K.H. Chua, "Integrated Prefabrication Configuration and Component Grouping for Resource Optimization of Precast Production", *Submitted to Journal of Construction Engineering and Management*, ASCE, 2012.

Conference Papers

- A. Khalili and D.K.H. Chua, "Precast Production Optimization using Prefabrication Configuration and Component Group concepts", EPPM Conference, Singapore 2011.
- A. Khalili and D.K.H. Chua, "Framework for an IFC-based tool for implementing Design for Deconstruction (DfD)", Computing Workshop ASCE, Florida, USA 2011.
- A. Khalili and D.K.H. Chua, "An IFC-Based Framework for Optimizing Prefabrication Configurations", Twelfth International Conference on Civil, Structural and Environmental Engineering Computing, Portugal 2009.